Is Real-time Phishing Eliminated with FIDO?

Social Engineering Downgrade Attacks against FIDO Protocols

Enis Ulqinaku,[†] Hala Assal,[*] AbdelRahman Abdou,[*] Sonia Chiasson,[*] and Srdjan Čapkun[†]
[†]*ETH Zürich, Switzerland, and* [*]*Carleton University, Ottawa, Canada*

Abstract

FIDO's Universal-2-Factor (U2F) is a web-authentication mechanism designed to provide resilience to *real-time phishing*—a class of attacks that undermines multi-factor authentication by allowing an attacker to relay second-factor one-time tokens from the victim user to the legitimate website in real-time. A U2F dongle is simple to use, and is designed to ensure users have complete mental models of proper usage. We show that social engineering attacks allow an adversary to downgrade FIDO's U2F to alternative authentication mechanisms. Websites allow such alternatives to handle dongle malfunction or loss. All FIDO-supporting wesbites in Alexa's top 100 allow choosing alternatives to FIDO, and are thus vulnerable to real-time phishing attacks. We crafted a phishing website that mimics Google login's page and implements a FIDO-downgrade attack. We then ran a carefully-designed user study to test the effect on users. We found that, while registering FIDO as their second authentication factor, 55 % of participants fell for real-time phishing, and another 35% would potentially be susceptible to the attack in practice.

1 Introduction

Fast IDentity Online (FIDO) is driven by an industry alliance with the goal of reinforcing web authentication by *"reducing the world's over-reliance on passwords"* [5]. The alliance has grown over time, now comprising 42 members including Amazon, Apple, Arm, Facebook, Google, Microsoft, PayPal, as well as financial corporations like American Express, Mastercard, Visa, and Wells Fargo.

The FIDO U2F standard defines cryptographic challenge-response protocols by which a hardware token (*dongle* henceforth) with a pre-baked cryptographic private key can prove its identity to a pre-registered website. The dongle interacts with a user's device through a Universal Serial Bus (USB) port, or wirelessly using Near-Field Communication (NFC) or Bluetooth (BLE). Such dongles are now manufactured by many companies, including Yubico and Feitian Technologies.

FIDO's U2F provides a high degree of resistance to exposing the secret key, comparable to some Physically Unclonable Function (PUF) technologies [3]. The challenge-response computations are performed on the dongle itself, and the private key never leaves the dongle. U2F thus enjoys relatively high resistance to the common cases of malware that runs on the user's machine. Physical theft of the dongle compromises its defence, however, such attacks are not scalable and cannot be performed remotely.

An important strength of U2F is that the domain (string) in the browser's address bar is a function of the challenge-response protocol. The browser[1] sends that string to the dongle. In the case of phishing [73, p.269], such domain string will be that of the attacker's website. Thus, an attacker relaying the result of the challenge-response from the browser to the legitimate website does not gain access because the response will not match the website's expectation. U2F is therefore a strong defender against phishing attacks [52], including the devastating *real-time phishing* attacks that undermine various Two-factor Authentication (2FA) alternatives [45, 49]. In real-time phishing, attackers relay the One-Time Password (OTP) (generated on the user's phone or sent over SMS) on the fly to the legitimate website. The FIDO alliance's webpage emphasizes the importance of handling phishing, and highlights the abilities of its suite of technologies in achieving this goal [4]: *"This security model eliminates the risks of phishing, all forms of password theft and replay attacks"*, *"[the] built-in phishing resistance and ease-of-use give it the potential to drive widespread adoption"*.

We show that FIDO could nonetheless be downgraded to weaker alternatives, enabled mostly by websites that allow users to choose second-factor alternatives to FIDO. Such alternative are typically provided to account for situations like U2F dongle loss, malfunction, or other reasons where a user simply wants to avoid using the dongle (*e.g.,* grant access to a remote spouse). Despite extensive design efforts to empower users with a complete mental model, and previous

[1] A compromised browser or a malware on the user's machine could send arbitrary data to the dongle, but they are outside the scope of FIDO.

literature showing the high usability and likeability that FIDO enjoys [28], we submit herein that clever social engineering tactics can enable a real-time phishing attacker to impersonate FIDO users, requiring neither malware nor dongle theft.

Consider for example a real-time phishing adversary. When the legitimate website prompts the adversary to insert the U2F dongle, the adversary likewise prompts the user on their phishing website. While the user inserts their dongle, the adversary asks the legitimate website to use an alternative method, and prompts the user to submit the OTP of that method on its phishing website. Users can perceive this as an additional third authentication factor, on top of the second factor—the dongle they just inserted, thus even higher security [46,61,77]. On the phishing website, the adversary simply ignores the response from the dongle, and relays the user-submitted OTP of the alternative method to the legitimate website, hence gaining access.

We manually inspected Alexa's Top 100 websites to verify if they allow choosing alternatives to FIDO during login. We found that all websites that support FIDO (23 out of 100) allow choosing weaker alternatives, therefore their users remain vulnerable to real-time phishing despite using FIDO. Ironically, most of these websites actually *force* users to first register an alternative 2FA method before being able to register FIDO as a second factor.

In this paper we approach two research questions. *(1) How susceptible are users to phishing attacks when using FIDO? (2) How do users detect phishing attacks when using FIDO?* By implementing a website that mimics real-time phishing of Google's login form, and through a carefully-designed user study of 51 participants, we found that only 10% of participants are unlikely to fall for (general) phishing in practice. They detected our phishing attempts early in the study, *e.g.,* from the phishing email or the phishing URL, before reaching our downgrading FIDO part. Had they missed the regular phishing indicators, it is unclear whether these participants would fall for the downgrade attack in practice. We found that participants did not change their phishing-detection techniques while using FIDO, which raises new questions as to whether users truly understand how the technology protects their accounts.

Contributions. This paper contributes:

- New social engineering attacks that allow an adversary to downgrade FIDO to weaker 2FA alternatives. Such alternatives are vulnerable to real-time phishing, which is the primary attack that FIDO protocols are designed to protect against. By allowing such downgrade, FIDO's defence against real-time phishing is rendered futile.
- New social-engineering evaluation methodology. Our evaluation methodology is designed to account for many aspects that typically negatively influence the results of a user study that evaluates attacks. We encourage its adoption in future research to mitigate biases.

Finally, none of the attacks discussed herein exploit weaknesses in the FIDO standards, APIs, or cryptographic protocols themselves. The core enabler is rather the availability of alternative authentication schemes. So long as users are allowed to login using weaker alternatives, attackers can, theoretically, always do likewise. However, with hardware tokens in general, it is necessary to either allow alternative login methods, or implement non-weaker account recovery mechanisms to account for token losses/malfunctions. Manual recovery is costly [53]. And with adversaries now capitalizing on an ongoing pandemic [39], and a global work-from-home pattern, it becomes increasingly important to make sure promising defences like FIDO are not undermined. Detailed discussion on countermeasures is provided in Sec. 7.

2 Background

In this section we review common 2FA schemes, real-time phishing attacks, and the FIDO specifications.

2.1 Two-Factor Authentication

2FA is a widely deployed strategy to strengthen password authentication. It usually requires users to enroll a second factor (*e.g.,* smartphone or special hardware) to their accounts during registration. Afterwards, upon submitting the correct password for login, the user is asked to prove possession of the second factor. To do so, most 2FA schemes require the user to submit an OTP displayed, or confirm a prompt, on their second factor [44,45,49,50,72]. To enhance user experience (reduce inconvenience of a method) and availability (access to the user's account), online services typically allow users to enroll more than one 2FA alternative per account.

Threat Model: Real-time phishing. Existing 2FAs protect users from password compromise but they largely remain vulnerable to *real-time phishing*. In real-time phishing, the user interacts with the malicious page (*e.g.,* googla.com) posing as the genuine website (*e.g.,* google.com), while the adversary authenticates simultaneously on the real website by relaying victim's credentials. The attack is relatively easy from a technical perspective, but very effective in practice [19]. Also, it is very challenging to be prevented because it mostly exploits human mistakes. Prompt notifications offer enhanced user experience, however, they put the burden onto users to detect ongoing attacks and risk user habituation [7]. Automated tools, *e.g.,* Evilginx [32], make *real-time phishing* easy to deploy and largely scalabe.

2.2 FIDO Specifications

Fast Identity Online (FIDO) is an open industry alliance that aims to reduce the reliance of web security on the user passwords, while preserving the usability. Major browser vendors

and service providers are part of the alliance and they are committed to implement and deploy the new specifications.

FIDO reduces the dependence on the user to detect phishing websites by assuming three trusted and cooperating components: i) *relaying party* is the server where the user authenticates; ii) *user client* is typically the browser that communicates the data received from the server together with the domain of the visited website to the authenticator; iii) *authenticator* is a device the user possesses, which stores a private key and signs messages received from the client. The key advantage of FIDO compared to other 2FA schemes is that the browser provides the authenticator with the domain of the visited website. Therefore, if the user falls for phishing, the browser communicates the malicious domain to the authenticator, which signs a message that is invalid to the honest server (because of the domain mismatch). Figure 1 shows an abstract challenge-response interaction.

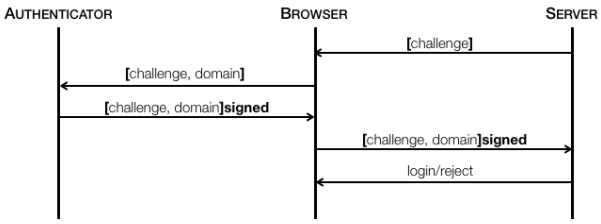


Figure 1: Information flow during an authentication attempt with FIDO. We assume the user has enrolled its authenticator dongle previously, so the server has the respective public key.

At the time of this writing, the alliance has published three sets of specifications [5] for secure user authentication: (1) *FIDO U2F* covers use cases where the authenticator is used as a second factor during authentication, *i.e.,* the user still has to type the username and password; (2) *FIDO UAF* known as "passwordless authentication" because the user authenticates only with one factor – the FIDO dongle. These specifications address also the authentication of the user to the dongle with a PIN or biometrics; (3) *FIDO2* which refers to the latest specifications published by the alliance and covers the use cases of both *U2F* and *UAF*. Unless specified, FIDO specifications refers to all three sets described above in the rest of this paper.

FIDO2 includes a web authentication API known as WebAuthn and the Client to Authenticator Protocol (CTAP2). CTAP2 allows abstracting the authenticator device and its implementation, therefore extends the set of devices that can be used as authenticators (*e.g.,* smartphones, smart watches, smart cards). Additionally, CTAP2 triggers browsers to display a prompt window, which includes the domain name, when an website tries to communicate with the dongle. However, CTAP2 is backward compatible and supports *U2F* functionalities (which do not trigger the prompt). Hence, an attacker can use the latter to avoid the browser prompt, or even exploit it to their favor (see our implementation; Sec. 4.1).

3 Problem Statement

For a long time the security of user authentication has been (and still is) a tradeoff between usability and security. Lang *et al.* [48] evaluated the security keys implementing FIDO specifications and concluded that security keys lack the *easy-recovery-from-loss* usability property in the UDS framework [12]. FIDO specifications focus mainly on the security of authentication with security keys, but provide only general recommendations for recovery [29].

Similar to authentication, account recovery is difficult. It directly affects the security of online accounts. Unfortunately, no known recovery mechanism exists that is efficient, secure, and scales easily to millions of users [53]. Previous literature [11,68,69] have shown that recovery schemes, especially those based on knowledge questions, have significant weaknesses. Therefore, service providers usually allow users to enrol more than one 2FA device with the assumption that users will have access to at least one when logging-in. This way, websites offer smooth user experience and reduce lockouts, which can cause financial loss to both the costumer and the provider [53]. However, except FIDO, none of the remaining second factors is secure against real-time phishing.

Reports from Google [20] and Microsoft [54] show that multiple 2FA schemes are widely deployed as alternative login mechanisms (user selects the 2FA challenge in every login attempt), or recovery mechanisms (user provides proof of 2FA to regain access to their account). Gelernter *et al.* [27] demonstrated that social engineering attacks on recovery methods are practical and effective. However, previous work on FIDO mostly focused on usability [16,23,28,31,65]; limited work questioned its security in real-world deployments, where secure alternative 2FA and secure recovery are necessary.

To measure the extent by which weaker 2FAs schemes are being offered as alternatives to FIDO in the real-world, we manually inspected Alexa's top 100 websites. We reviewed documentation (when available) for websites' policy regarding authentication when FIDO was supported, and created accounts on those websites that offer public access to test their policy in practice. Results are shown in Table 1. All 23 websites (belonging to 10 organizations) allow choosing alternatives to FIDO. Users of these sites thus remain vulnerable to real-time phishing, despite enabling FIDO. More disturbing, most of these websites *force* users to first register an alternative 2FA before being permitted to enrol their FIDO dongle, which essentially undermines the added security of FIDO.

4 Downgrading FIDO via Social Engineering

The new downgrade attack on FIDO presented herein is a type of real-time phishing (Sec. 2). The attack starts as a typical

Table 1: All 23 websites in Alexa's top 100 allow weaker 2FA alternatives to be registered alongside FIDO.

	Support FIDO		Do not support FIDO	Total
	allow alternatives	do not allow alternatives		
FIDO partner	14	0	15	29
Others	9	0	62	71
Total	23	0	77	100

real-time phishing (see Fig. 2), with the user on the phishing website and the attacker on the legitimate website at the same time. After relaying the user's credentials (Step 2 in Fig. 2), the attacker is presented with the FIDO-prompt page from the legitimate site (Step 3), and in turn displays a FIDO-prompt page to the user (Step 4).

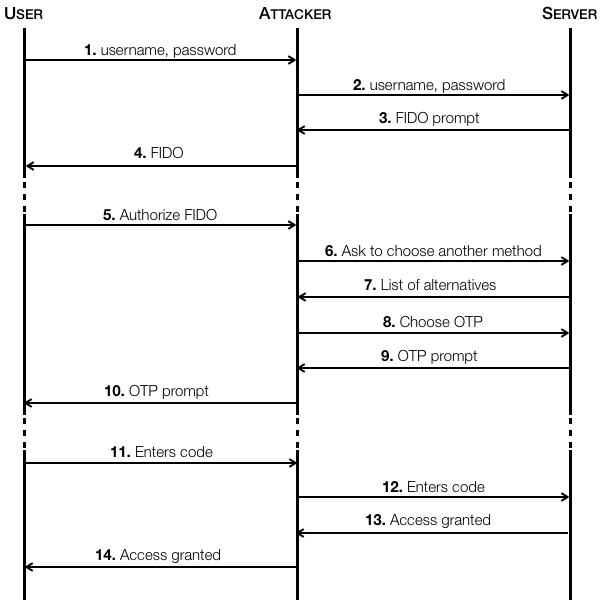


Figure 2: Downgrading FIDO via social engineering. Dashes indicate longer time stretches, reflecting when the user acts.

At this point, the attacker waits until the user authorizes their FIDO token to interact with the attacker's page through the browser (Step 5).[2] The attacker can leverage standard API functions (*e.g.,* `u2f.register` and `u2f.sign` for U2F), so that the attacker is notified when such an authorization-for-interaction occurs. When the browser communicates the result of the challenge-response, the attacker ignores the result of this interaction because all they need to know is that the user has inserted the token. The attacker then chooses, on the legitimate website, to use an alternative second factor method from the list pre-configured by the (victim) user on the website (Steps 6–9), and displays a page prompting the user for that same method (Step 10). Depending on the website, this step can simply be presented to the user without any indication as to whether her FIDO-trial was successful. In our phishing implementation below (Sec. 4.1), we show how Google's default message to users helps our (attacker's) cause.Upon getting the token from the user (Step 11), the attacker forwards it on to the legitimate website (Step 12), hence gaining access.

Timing and ordering notes. In Fig. 2, Steps 6–9 can vary between websites; some present the user with options; others may choose for the user. These four steps (*i.e.,* 6–9) must however occur quickly so that the page in Step 10 is displayed to the user right after the user's FIDO authorization in Step 5. To speed-up displaying the OTP prompt to the user (Step 10), the attacker can initiate Steps 6–9 before 5, so that the OTP prompt (Step 10) is ready immediately after the user's authorization. However, the delay between Steps 9 and 12 must also be kept small before the website's OTP token expires (if it was SMS; for an app-generated OTP, the attacker only needs maintain a small delay between Steps 11 and 12). All such steps can be automated, thus delays can be kept minimal.

Reflections on Step 10. A key element in this attack occurs in Step 10, where the user will be prompted for another authentication factor after using the FIDO token. Seeing three login methods (password + FIDO + OTP) likely sends a false signal to the user that this login trial is even more secure than with only two factors (password + FIDO). Google's OTP page, for example, has the sentence *"This extra step shows that it's really you trying to sign in"*. When an attacker displays that page after its fake FIDO-prompt, the user would interpret it as an "extra" beyond password + FIDO, but it is intended (by the legitimate site) as extra to only the password. In our implementation (below), we constructed this (phishing) page with the statement as-is. The "extra step" here *enables* our attack, as it helps attackers downgrade FIDO to other methods.

Variations to Step 10. Depending on the design of the legitimate website, variations other than presenting a page with an alternative authentication (Step 10) immediately after the FIDO prompt may be more effective in tricking the user. For example, the attacker may display: *"due to technical error, we are unable to process your FIDO token at this time"*, or *"our FIDO-handling service is currently down, please use another method"*. The latter avoids the use of FIDO APIs altogether, so alert messages familiar to the user in the browser-displayed FIDO-prompt box (where attackers have no control over the message within) are avoided.

4.1 Attack Implementation

In preparation for running a user study to test the effectiveness of this attack, we implemented a phishing website that behaves in the manner explained above. The website targets Google's login page. Details of the user study, including ethical considerations, are discussed in Sec. 5.

Our phishing pages are shown in Appendix A (Fig. 7). We obtained the domain `two-step.online` as our phishing

[2] Some models require a button press; others a touch.

domain, got a Let's Encrypt certificate for the domain, and placed our phishing pages inside a `google.com` directory on our server. We intentionally opted for a domain with valid words in a non-traditional TLD such as `.online` for two reasons: (1) we could get a TLS certificate without being flagged as suspicious [64], and (2) users that do not understand how URLs work but might have a look at it would not be alerted as `google.com` is present [78]. The `index.html` page would get periodically blocklisted [3] every few days, and so we hid it such that it is only accessible through a 41-character randomly generated alpha-numeric value stored in a variable that we called `acc`. The page would thus only be reachable by a link, which would be emailed to potential victims. While on the phishing website, the browser's URL bar would have a nice green padlock icon with the URL:

https://login.two-step.online/google.com/index.php?acc=8[..]b

Corresponding PHP code at the start of `index.php` reads:

```
<?php header('Access-Control-Allow-Origin: *');
if (htmlspecialchars($_GET["acc"]) != "8FkuX...") {
  echo "This is Index.php!"; exit(0);
  } ?>
```

When implementing our phishing pages, we did not borrow content from Google's website; we neither pre-downloaded content from Google to upload to our pages, nor linked to Google content from our pages. The former is not quite straightforward because Google employs code obfuscation techniques on its webpages (*e.g.,* to thwart phishing attacks); the latter was avoided to evade potential phishing detection through analyzing our server's requests to Google's web-content [59]. The only object we downloaded and uploaded onto our server was Google's logo (image). Note that creating our phishing page would be feasible for any attacker with moderate web programming experience. Our implementation of Google's pages resulted in fewer than 2K lines of combined PHP/JavaScript/HTML/CSS code.

Recall from Sec. 2, the authentic FIDO prompt is typically displayed outside of the attacker-controlled area of the browser to prevent attackers from replicating the prompt within the content pane; note, *e.g.,* for Chrome, the top tip of the box overlapping the URL bar (see Fig. 6c in Appendix A). Recall also that browsers capture the domain from the URL bar and display it to the user within the FIDO-prompt box. It is thus helpful (to the attacker) to use API functions that do not display this box to the user, yet gets the browser to notify the webserver that a dongle was inserted. For Step 10 (Fig. 2), we used the `u2f.register` function, which does not display browser-generated prompts. With this function, communications with the user are left to the website developer (*i.e.,* through standard HTML and JavaScript).[4] As an attacker, we do not control the legitimate displayed message; it is browser-generated. So we implemented a mimicry of the Chrome-generated FIDO prompt as a `gif` image that looks like Chrome's box, with a message identical to the authentic one: *"Use your security key with google.com"* (Fig. 7c, Appendix A). The `gif` had an animated indeterminate progress bar, almost similar (visually) to Chrome's authentic one (Fig. 6c, Appendix A). Since it was an image, it was fully contained within the browser's content pane, located vertically at pixel 0 (top-most point).

Finally, since our aim is only to test the effect of our attack on participants in Sec. 5 (*i.e.,* we do not want to actually steal credentials), we did not implement back-end communication between our phishing website and Google's site. We rather replicated relevant login functionality, including allowing usernames to be entered with and without the *@gmail.com* suffix, and ignoring dots throughout the username.[5] Our phishing site also handled situations outside of the normal phishing path, such as a non-existent username by displaying an error page similar to Google's (Fig. 8, Appendix A). We allowed only the username used in our user study (below).

5 Evaluation Methodology

We designed a user study to test the effectiveness of the above social engineering tactics. In comparison to studies that test the usability of systems, designing a user study to test attack effectiveness is often challenging. The study must be ethical. It should reflect a user's true keenness in protecting their assets. Moreover, the explanation of the study tasks to participants should not (1) artificially lead participants to fall for the attacks in question, and (2) artificially alert participants so they detect/avoid the attacks.

For example, evaluating the effectiveness of a phishing attack using the participants' real accounts (*e.g.,* personal email) in the study is high risk, even if the phishing website and the study were configured such that it is impossible for any of the researchers to access the stolen credentials. On the other hand, participants may be careless with the credentials of accounts created just for the study. If asked to login to a series of websites (while burying phishing websites amongst these), a participant may proceed to login on all sites even if phishing was noticed, thinking that doing so is part of the study instructions; if asked to avoid bad sites, a participant can become more cautions than they would in practice.

We now explain how we designed our user study to minimize the effects arising from the above challenges.

[3] Alternative to the term 'blacklisted' as per USENIX's recent commitment to inclusion efforts (https://bit.ly/31cNJib)

[4] Note that even if a browser-generated box was used, users may already be oblivious to the messages displayed within that box.

[5] The dot is discarded in Google's user accounts. That is, *john.smith@gmail.com* is the same as *johnsmith@gmail.com*

5.1 Study Design

The study advertisement generically explained that the purpose of the study was to evaluate and improve the usability of email clients. To eliminate any later doubt by participants about the safety of their legitimate credentials, participants did not use their own email accounts. We provided user accounts and credentials created specifically for this study. However, to maintain ecological validity, we designed a study scenario that indirectly encouraged participants to think about the security of these accounts.

We ran the study concurrently in two cities, one in North America, below suffixed with *-N*, and one in Europe, *-E* (Ottawa and Zürich respectively). To maintain consistency in both cities, we carefully documented the study protocol and had the two researchers running study sessions follow this common protocol. Participants were monetarily compensated for their time, $10 in Ottawa and CHF20 in Zürich. Participants first completed a demographics questionnaire then they went through the study scenario, during which they were asked to *think-aloud* (*i.e.,* to describe their thought process out loud). We next gathered feedback from participants through a semi-structured interview. At the session's end, the researcher provided participants with a debriefing form, explaining the true purpose of the study and answered any questions they had. Study sessions were audio-recorded, and the interview portion was transcribed for analysis. The study received IRB approval in both cities.

5.1.1 Study Scenario

Participants were asked to role play *Jordan Hart*, a new employee in a technology company on her/his first day of work. They were provided with their company gear: a laptop, smartphone, and a security key (the FIDO U2F dongle). Participants were asked to read and sign the employee on-boarding information sheet (Appendix C), a common practice in industry. This sheet outlined the company policy with respect to safeguarding company information and avoiding scams and phishing attacks, as well as explaining FIDO keys and their associated security benefits in language adapted from Google's *Security and identity products* pages [31]. The sheet listed Microsoft Outlook as the company's primary email provider, included Jordan's Outlook account credentials (username and password), as well as provided their Google services credentials. We created real Microsoft and Google accounts. The sheet also included the names and email addresses of Jordan's manager, IT manager, and HR person, from whom Jordan would receive emails. We created real Microsoft email accounts for each of them. To make sure participants were comfortable using the FIDO key, the researcher–acting as the IT manager–asked participants to login to their email account with the key as a second factor, and explained how to use the Google Authenticator app (pre-installed on Jordan's smartphone, and configured for use on Jordan's Google account) in case of any technical difficulties.

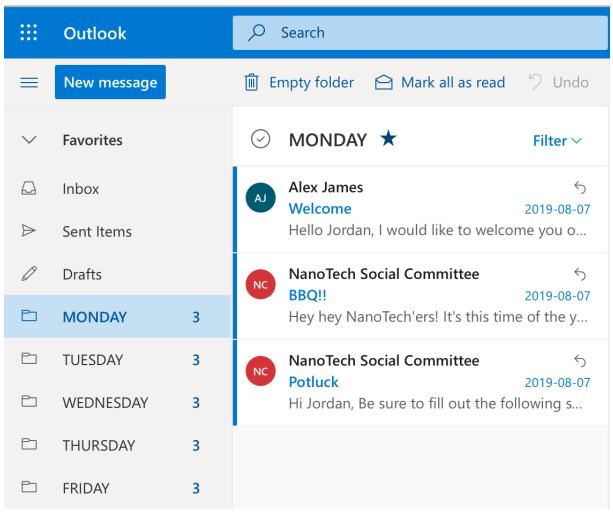


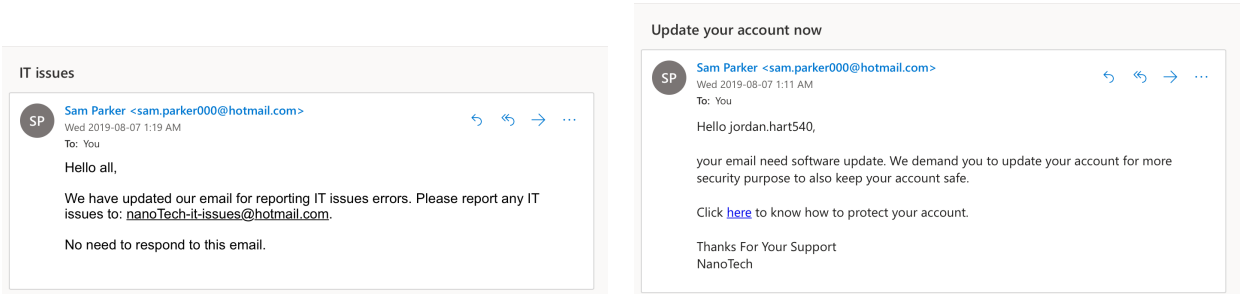
Figure 3: Emails were divided into 5 daily folders. Monday emails are shown here.

Jordan's Microsoft email inbox contained 15 emails, divided into 5 folders, one for each day of the week (Fig. 3). Participants were asked to assume that they login to their Outlook account daily, handle emails received that day (as tagged), logout and shutdown their laptop before going home, and come back the next day to do the same steps. The researcher simulated shutting down the laptop when indicated by the participant by logging-out of their email and clearing the browser cache after finishing each day's emails. Participants used the Google Chrome web browser.

5.1.2 Emails

Four of the 15 emails were phishing, containing a link to our phishing website (Sec. 4.1). Such emails were spearphishing (targeted). We used PHP's `mail` function to send out these emails using a spoofed source email address. To ensure realism, the phishing emails included errors like grammatical mistakes and typos, mimicking typical phishing emails. Non-phishing emails were sent from the authentic email accounts of the companies employees (Jordan's manager, IT manager, and HR person) through the email web client. All emails were sent once before we started recruiting participants, and simply marked as unread before the next participant. When we initially sent them, we manually moved those that were placed into Jordan's Spam folder (legitimate or phishing) into the Inbox folder. Figure 4 shows a legitimate and a phishing email, both appearing to be from the IT manager. Note that, as in real-life, when visiting our phishing pages, participants will see the fake login form even when they are already logged-in to Jordan's Google account. This has alerted vigilant participant, P2-N, to our phishing attempts.

Some emails, legitimate and phishing, included links to doc-



(a) Legitimate email

(b) Phishing email (address spoofed)

Figure 4: A phishing and a legitimate email that appear to be from the same sender.

uments. We created actual documents for every such email, and stored them on Google drive. Legitimate documents were only accessible through Jordan's Google drive account. We (attacker) set the other documents on Google drive as accessible with a link, and redirected to them after the user finished logging-in to our phishing website. This way, the browser's URL bar would display an authentic Google domain after the participant's persona credentials were phished.

5.2 Participants

We recruited 51 participants for this study: 25 in Europe and 26 in North America. Our dataset is balanced in terms of gender: 26 participants identified as female, 24 as male, and one chose "Other or prefer not to answer". The vast majority of participants had an undergraduate or a graduate degree ($n = 46$). Appendix B summarizes participants' demographics.

6 Results

We used the Qualitative Content Analysis Methodology [43] to analyze qualitative data collected throughout the study (*e.g.,* post-testing interview scripts, and researchers' notes). We developed an analysis matrix to cover the main topics relevant to our research questions. The matrix comprised of four categories with which we coded our data: *identifying phishing links, participants' perception of FIDO, their perception of 2FA, and their security attitude and awareness*. We then followed an inductive analysis method, and performed open coding to look for interesting themes and common patterns in the data. Themes irrelevant to our research questions are not discussed herein. A single researcher coded the data, however two researchers met regularly to discuss the codes and interpret the data. We followed recommendations by previous work [14] to have a single coder with considerable experience in the domain, so that this researcher would perform rigorous analysis by being immersed in the data. Through data analysis, we intended to see if there would be differences in the results between the North American and European cities, where the studies were conducted. We found no clear distinctions. We thus discuss the amalgamated results, within the context of the two research questions in Sec. 1.

6.1 Phishing susceptibility with FIDO (RQ1)

To identify participants who could be victims to our attack in practice, we need a mapping between their behaviour in the study and their attack susceptibility in practice. Simply classifying those who have submitted their credentials to one of our phishing links as potential victims may not be accurate because: (1) participants may not be as keen to protect their study credentials as they would their own, and (2) participants may still think they need to process all emails regardless of their suspicion because this is what the study is asking them to do. As explained in Section 5.1, we took measures to reduce the impact of both points, *e.g.,* through emphasizing the importance of security to the persona's employer, and integrating actions with interview responses. Only two participants mentioned they were not paying attention because they thought it was what the study asked them to do, highlighting the importance of our measures.

We also want to determine if participants were aware of phishing attempts in the study. Asking participants about each email, one-by-one, whether it was a phishing attempt risks making them overly vigilant, and potentially biased to answer "yes". Instead, we designed the post-study interview such that we *indirectly* gauge participants' awareness of the phishing attempts. Following previously established notions of determining participants' thoughts [27], we asked:

"If we told you that 50% of our participants access fake websites during their study sessions, do you think you are one of them? Why? Why not?"

We still allowed participants to go back to the emails and check them during the interview, should they ask to do so.

Participants' actions (during the study) and awareness of the attacks may or may not align. Four possible outcomes

emerge for each participant, summarized in Table 2. We classify those who noticed phishing attempts during the study as *aware-of-phishing-attempts* in the table. Normally, a participant who was unaware of our phishing attempts would submit their credentials to the phishing website. This is Case 1 in the table. A vigilant participant would normally refrain from submitting their credentials, and confirm their awareness of phishing attempts in the post-study interview—Case 4.

Cases 1 and 4 are straightforward; we classify the former as "susceptible to phishing", the latter as not. We classify Cases 2 and 3 as "potentially susceptible to phishing". In Case 2, although they did not submit credentials, participants were unaware of any phishing attempt. In Case 3, participants were classified as aware, yet they submitted credentials.

6.1.1 Participants' awareness of our phishing attempts

Determining participants' awareness from the interview is not trivial. Participants responses' varied substantially. For example, to the above question (*"If we told you that 50%..."*), some participants gave an affirmative response, but only name examples of non-phishing emails. Others answered affirmatively, but said they did not remember which ones were phishing. We also had participants who first denied being in the 50% that accessed fake sites, then hesitated, alternating between "yes" and "no", then changed their minds, and gave a few true phishing examples. And there were participants that provided an immediate affirmative response, reconsidered, and finally decided there were no phishing emails.

We thus ignored their direct 'yes/no/maybe' response and instead relied on the more objective portions of their comments to assess awareness, as described next.

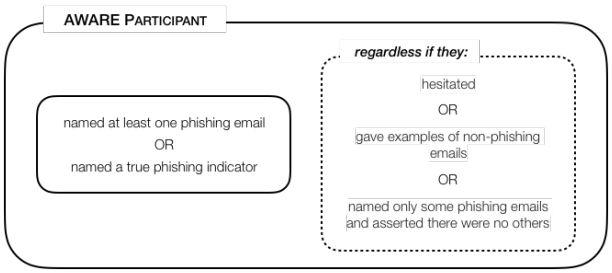


Figure 5: Determining awareness of our phishing attempts.

Any participant who (i) identified at least one phishing email or (ii) named a true phishing indicator is classified as *aware-of-phishing-attempts*, regardless of what else was said during the interview. Figure 5 shows this criteria, alongside common example responses in our study that we discarded because the awareness criteria was met. By *true phishing indicator*, we mean the website's URL, and commonly agreed upon (though non-robust) signs of phishing emails [38], like typos, lack of context, and grammatical mistakes. Unencrypted email is an example of *false phishing indicators*.

Table 2: Classifying participants' susceptibility to our phishing attack in practice, from their study behaviour. A checkmark (✓) under "submitted credentials" refers to a participant who has submitted their login credentials to at least one of the four phishing emails in the study.

Case	Participant		Susceptible	Results	
	aware-of-phishing-attempts	submitted credentials		#	%
1	✗	✓	Yes	28	55
2	✗	✗	Potentially	1	2
3	✓	✓	Potentially	17	33
4	✓	✗	No	5	10

From the post-study interview, we found that participants with responses that did match our criteria were doubtlessly unaware of our phishing attempts. This includes participants that: just denied being in the 50%; affirmed being in the 50%, but gave only examples of non-phishing emails; and affirmed but gave only false phishing indicators.

Conservative classification of attack awareness. Following the above criteria (Fig. 5), we classified participants as *aware-of-phishing-attempts* even in situations where it is hard to tell whether they were truly aware of such attempts. Thus we provide an upper bound on awareness. For example, a participant who named a true phishing indicator, yet asserted seeing no phishing emails is still classified as *aware-of-phishing-attempts*. Classified likewise is a participant who, *e.g.,* gave an example of one phishing email, mistakenly identified two non-phishing emails, and asserted there were no other phishing emails (*i.e.,* missed the three other phishing emails). We used conservative criteria for two reasons: (1) we increase certainty that participants classified as *unaware-of-phishing-attempts* would most likely be unaware of similar attempts in practice, and (2) participants may have forgotten which emails were truly phishing by the time they reach the post-study interview, as there were 15 emails in total. We purposefully avoided showing each of the 15 emails to participants and asking them which were phishing to avoid priming. Our hypothesis here is that, if during the study, a participant suspected a phishing attempt, they would recall that and indicate it in a manner captured by the criteria in Fig. 5.

Conservative classification of susceptibility to attacks. Our determination of susceptibility is based on two factors that we first assessed independently: awareness and submission of credentials. We classify only the most blatant cases as truly susceptible, again opting for a conservative estimate and thus providing a lower bound for susceptibility. Our aware/unaware classification is not per email, but rather per participant. So even if a participant named one phishing email in the interview but missed all others (or asserted there were no others), they will still be classified as *aware-of-phishing-attempts*. When we check whether this participant submitted credentials to our phishing website, we do not match the phishing

email(s) they detected (or fell for) in the study with what we classified as *aware-of-phishing-attempts* (or unaware) from the interview. For example, a participant who noticed only one phishing email, E2, is classified as *aware-of-phishing-attempts*, even if they asserted there were no others; if this participant submits credentials upon clicking on the link in phishing email E3, we classify them as "potentially susceptible", not as "susceptible". One would argue that this is a "susceptible" participant because there is an email that successfully phished the participant's credential, and it was clear that the participant was unaware of it. Being conservative, we opt to use any minor indication that a participant might notice similar attacks in practice as grounds for avoiding classification as "susceptible".

Examples of aware participants. In response to our interview question (*"If we told you that 50%..."*), the following are examples classified as *aware-of-phishing-attempts*. P17-E said, *"No, I think I haven't... Ah! maybe this Sam Logan is a phishing [email]. [...] he [emailed] twice, it could be... I don't know. If I got phishing, this is the only email I feel it could be."*. P17-N said, *"Yes, [I was in the 50%] [...] I was taking it for granted that the emails I was getting from the employees at the company were legitimate. [...] So I think that Sam Logan ones were, at least the one that I got from Sam Logan on the Friday was definitely a phishing email [...] Now that I'm thinking about it, that was definitely a phishing email, because of how poorly worded it was."*. P25-E said, *"I received many phishing emails here (identified them correctly during the study). I think there were two types, first the email about account change. The address looked it is coming from the source but as the company doesn't have any encryption I cannot be sure. I would have gone physically to the person. And the others that asked for google credentials, for those I just checked the address."*

Examples of unaware. P10-N said, *"I don't think so. [...] Everything seemed legitimate enough and seemed business-y. And I look[ed], everything looks like pretty work-related and exactly related to what the e-mail said it would be. Yeah. It wasn't like I just clicked on a link and it really brought me to some random page or something, it was related to what the e-mail was saying. So it seems legitimate to me."*. P13-N said, *"I just went to hotmail, the outlook website which I very often go. And I logged in from there. So I think it seemed fine."*

6.1.2 Susceptibility to Phishing: Results

Table 2 summarizes the results; 57% of participants were classified as "unaware of phishing attempts", and only one of those did not submit credentials to our phishing website. As such, given our conservative measures in classifying susceptibility, our results suggest that at least 55 of participants would be susceptible to our phishing attacks in practice. In contrast, 43% of participants were classified as aware of phishing attempts, and 23% of these did not submit credentials to our phishing website; at most 10% of our participants are likely to detect the discussed phishing attempts in practice.

The one participant in Case 2, P12-E, was very rapid in going over the emails. She did not click on any phishing link, and haven't also clicked on several non-phishing links. She gave very short, non-informative, responses in the post-study interview. When asked why she did not click on links in the emails, she simply said, *"There is no particular reason"*.

Takeaway. Our focus in the present paper is to determine user's susceptibility to phishing, particularly while using FIDO. We noticed that all participants who appear to have detected and avoided our phishing attempts (Case 4) would have done so also without using FIDO. The phishing indicators they mentioned, and the reasons they discussed as to why they avoided submitting credentials to our phising site are not related to FIDO. Likewise, those whom we classified as susceptible to phishing are susceptible despite using FIDO. That is, using FIDO did not protect them from our downgrade attacks. Essentially, what we were looking for in this research is cases of users who would have fallen for phishing without FIDO, but have not because of using FIDO. We found none.

6.2 Phishing detection while using FIDO (RQ2)

In the post-testing questionnaire, we asked participants if they had accessed fake web-sites during their study session, and we found that this question evoked participants to think about the emails more deeply, and discuss signs for phishing attacks. Through our analysis of the qualitative data, we identified seven phishing attack indicators summarized in Table 3. These indicators show that participants identify phishing attacks when using FIDO just as they would without using it. Participants discussed common advice for identifying phishing emails (*e.g.,* the presence of a hyperlink, the email is out of context, and the the tone of the email is inappropriate). Three participants also discussed that *repeated login* prompts was an unusual behaviour that seemed suspicious. P2-N explains, *"It makes me want to log onto Google even though I was already logged on to Google on just another tab.[...] This was not a thing I noticed at the beginning when I was doing the experiment [...]. Now that I'm thinking about it. Yeah, makes sense, right? Like why are they asking you to log onto Google again when you're already logged onto Google?!"*.

On the other hand, we explored participants' reasoning for assuming that they were not victims of a phishing attack in our study (see Table 3). Participants relied on some observational safety indicators, *e.g.,* the information is received through their company's official channels, the sender is a colleague whom they know, the context of the email is expected, and the login pages (for the email and Google Drive) looks legitimate. Two safety indicators relate to the webpages linked in

the email, opening these pages redirects participants to the expected content (*e.g.,* a Google Sheet), and does not lead to unexpected behaviour (*e.g.,* popups). Some participants ($n = 3$) indicated they *"felt more secure with 2FA"* (P23-E) and were protected against phishing because they were using FIDO. P20-E explains, *"I think after reading at the [new employee] information sheet I thought, oh it's safe, I don't have to worry about [phishing emails]"*.

Interestingly, requiring participants to use the Google Authenticator, which is part of our attack, gave some participants a false sense of security. P21-N says *"I had to put in the information [code] as well and I felt secure: the company even took me to verify everything [using the Google Authenticator] to make sure that it was secured"*. These participants either considered the authenticator an additional factor or assumed it was part of how FIDO works, and some even thought FIDO was more secure because of the authenticator. P10-E explains, *"If you have to use the authentication app on the phone, with the changing number always, it is really difficult for someone to hack your system to find this kind of information."*

Takeaway. Despite using FIDO, we noticed that none of the participants have relied, or indicated that they would rely, on FIDO for detecting phishing attempts. Evidenced by our attacks, the proper usage would be to refuse to login with alternative methods if a user has enabled FIDO. In contrast, we saw three participants who said they were secure because they used FIDO in all their logins, even when some of these were accompanied by other authentication factors. Seeing a FIDO-*only* login is practically opposite to using FIDO alongside other factors—the former prevents downgrade attacks, the latter enables them. We found no evidence that any of our 51 participants understood this concept.

7 Discussions and Countermeasures

Social engineering attacks are difficult to mitigate. They exploit human error, and trick users to follow malicious instructions, thus perform insecure actions unwittingly. Our user study is designed to evaluate the effectiveness of phishing attacks that downgrade FIDO to weaker alternatives. We provide practical insights, partly from our results, regarding potential defenses.

7.1 Disable Weaker Alternatives

A straightforward countermeasure to the downgrade attack presented herein is to disable alternative 2FA methods if a user enables FIDO. Google's advanced protection program [30] achieves this for critical accounts, *e.g.,* those of politicians or journalists. The program is opt-in and the participating users should register at least two security keys, one for daily use,[6] and others as backup. However, Google does not detail the recovery process in case both keys are unavailable, but states that *"it may take a few days to verify it's you and restore your access"*. This delay poses a major trade-off for users to choose between additional security against phishing versus the availability to access their accounts any time.

Limitation: non-scalable recovery. Doefler *et al.* [20] report that challenges requiring security keys have lower pass rate than device-based ones. So, if alternatives were disabled, more users would need to go through the recovery process. On the other hand, such recovery adds significant costs to service providers, and does not scale to millions of users [53]. Disabling weaker FIDO alternatives comes at the cost of non-scalable recovery.

Limitation: usability impact. Previous literature [16, 23, 65] reported that users have difficulties enrolling security keys into their accounts, and are concerned about being locked out in case keys are lost. Registering multiple keys can enhance the user experience but may be costly for users,[7] which might be a barrier to some users. Moreover, service providers tend to facilitate user onboarding and enhance overall experience by offering a variety of channels to connect to its backend, *e.g.,* browsers, native apps on different OSes, or third-party software such as email clients. Disabling FIDO alternatives can degrade usability because channels that do not support FIDO should then be dropped—otherwise, the attacker connects to the server through such channels. In summary, disabling weaker FIDO alternatives is complex because a provider should implement a scalable and secure recovery, and consider all channels used to connect to its services.

7.2 Risk Based Authentication

Risk-based Authentication (RBA) refers to a set of server-side techniques to assess the risk of an authentication attempt, and block malicious ones [33, 70, 75]. Secure IP geolocation [2], device, network, user agent, and installed plugins are examples of metadata that RBA systems analyze for deciding the risk score of a login attempt. A low risk attempt (*e.g.,* same user agent and same IP address) gives confidence to the server that the honest user is authenticating. For higher risk requests, the server challenges the user to provide additional factors, or restricts user's access depending on the provider's policy [76].

Limitation: mimicry of user's attributes/behaviour. A recent study [13] shows that attackers have already developed malicious tools that can circumvent RBA defenses. Such tools are made available as public services. Campobasso and Allodi [13] reveal that attackers collect necessary data from victims on top of their credentials, so they can bypass RBA

[6] A phone running Android 7+, or iOS 10+ with the Google Smart Lock app, can be used as one security key.

[7] At the time of this writing, security keys from Yubico (a popular vendor and FIDO Alliance partner) cost around $20.

Code	Explanation	Example Quote
Phishing attack indicators		
Context	The circumstances surrounding the email received and its subject; the timing of the email in terms of events is inappropriate/unexpected	*"there was [an email] that [was] for a job or something, and I was thinking I already have a job, I thought it was weird"* (P14-E)
Grammar and styling	The email contains mistakes in grammar, punctuation, or capitalization	*"Now that I'm thinking about it, that was definitely a phishing email. Because of how poorly worded it was."* (P17-N)
Hyperlink	The email includes a hyperlink	*"Um well, most of the red flags I got were from when there is a link in it."* (P7-N)
Repeated logins	The participant is required to login although they have already logged in and the session is supposed to be maintained	*"I logged in my Gmail, and then I clicked on an email again. And I had to, re-enter my login credentials. Like something like this ought to be kind of phishing"* (P15-N)
Sender language consistency	The language in the email is not consistent with how the sender usually writes emails	*"[That's] not the right person, that's not the person I know from the way, it's the tone of writing and the language and the way it's said."* (P20-N)
Tone	The tone of the email is unexpected (*e.g.,* demanding, or not professional as expected in the workplace), or the email does not include greetings or greets the receiver by their username rather than their name	*" 'We demand you' I feel like somebody would not be using that kind of language at work."* (P1-N) *"One email was not addressed to me with a name, but to the username, so it looked like a bot."* (P19-E)
URL	The URL of the hyperlink in the email is suspicious	*"The URL looks really weird, I think it's not safe, or like that's not the normal. This is just like fanciness that looks like Google"* (P11-E)
Reasons for safety		
Context	The circumstances surrounding the email received and its subject; the timing of the email in terms of events is appropriate/expected	*"If it is just, like my boss sending a book to download, and we talked about it, it's fine. But if it is a random book, then it's weird. [...] I think if [the download book email] was sent to me in real life, I would click on it, because it is mentioning nanotechnoloty, it has a context that makes sense"* (P16-E)
sender address	The sender's address is correct in the email header (The FROM part of the header)	*"I verified their email [address] and some like I would assume that, that is the legitimate person"* (P11-N)
Sender	The receiver knows the sender, the email is not from a complete stranger	*"Since this is a secure network, and all the people that were sending me emails were company, colleges, I suppose there were no phishing emails"* (P24-E)
URL	The URL of the hyperlink in the email looks legitimate	*"I didn't click any of the suspicious links. I mean, I did click links to Google Docs and things like that and they looked legit to me"* (P2-N)
communication channel	The emails and linked content were sent through the official company emails, by employees of the company	*"I didn't open something that looked suspicious. [...] Everything was from official channels, from work, so I think it should be ok."* (P10-E)
login interface	The login interface looked legitimate	*"I was logging in to the right thing. Most of the things that came up were Gmail and Outlook."* (P18-N)
popups	Clicking on the hyperlink the email did not lead to popups	*"I don't know that anything is entirely compromised but maybe I clicked on a link, but I didn't see any indicators of that. Like I didn't see like any pop ups or any extra spam come in or anything like that"* (P25-N)
content	The hyperlink in the email redirected the user to the expected content	*"Everything looks like pretty work related and exactly related to what the e-mail said it would be. Yeah. It wasn't like I just clicked on a link and it brought me to some random some random page or something, it was related to what the e-mail was saying. So it seems legitimate to me."* (P10-N)
using FIDO/2FA	Using FIDO/2FA makes it more secure	*"It kind of seemed to be fine, I suppose I felt more secure with with the 2FA [FIDO token] because they cannot steal all information if it is encrypted."* (P23-E)
antivirus	Relying on the antivirus to handle security	*"I am kind of a lazy person and as I said before I rely on my antivirus too much, but I guess it is what it is"* (P11-E)
Google authenticator	Requiring google authenticator is an added level of security	*"I had to put in the information [code] as well and I felt secure: the company even took me to verify everything [using the Google Authenticator] to make sure that it was secured"* (P21-N) *"More steps [authenticator + FIDO], more security"* (P13-E)

Table 3: Phishing indicators and Reasons for safety

defenses. Similarly, an adversary performing real-time phishing can adapt such tools to bypass RBA mechanisms on the fly. This adversary has a connection with the victim's browser, and may be able to mimic attributes/behaviours to the legitimate website [3], or execute the JavaScript code (related to RBA analysis) directly on the victim user's browser.

7.3 Browser Hints

The recent WebAuthn API [8] instructs browsers to always show a prompt window when a website interacts with the authenticator during both: registration and authentication. The prompt is part of the user consent, which means that the user agrees (by tapping the authenticator device) to complete the request displayed on the prompt. The prompt itself contains a short message, and browsers display it as a native popup that extends slightly above the address bar. For example, Google Chrome captures the TLD and second-level domains of the website (*e.g.,* `google.com`), and displays them to the user within the prompt box alongside the message:

> Use your security key with google.com

Mozilla Firefox includes the fully qualified domain name (*e.g.,* `accounts.google.com`) in a callout panel as:

> **accounts.google.com** wants to authenticate you using a registered security key. You can connect and authorize one now, or cancel.

Since the prompt contains a short message and the website's domain, rendered in boldface in Firefox, it can potentially alert visitors of a phishing website.

Limitation: users' susceptibility to social engineering. Relying on users to notice the domain mismatch should not

be part of the protocols' security for three reasons. First, FIDO promises to relieve users' from the burden of detecting phishing, hence security should not depend on prompts or visual indicators. Second, previous research [7, 19] have shown that users typically do not pay attention or understand browser hints related to security. Third, the adversary can use the U2F API to interact with the device, which does not trigger such prompt windows (as we did in our implementation—Sec. 4.1).

7.4 Secure Login and Recovery Alternatives

Doefler *et al.* [20] discuss Google's categorization of login, second factor authentication, and recovery methods. Methods of comparable security are placed in the same category, and should be allowed depending on the account's security status. Such a status could possibly be indicated by the user's security configuration (*e.g.,* enabled 2FA, configured robust recovery methods).

Promising Countermeasure. It appears that a viable countermeasure to the attacks discussed herein is: when FIDO is enabled, only enable authentication (or 2FA) alternatives that provide resilience to similar attacks that FIDO is designed to protect against. Suitable candidate alternatives include other FIDO protocols. For example, a phone-based authenticator through FIDO2 can serve as a suitable authentication alternative to physical security keys. This should be recommended/enforced by service providers (websites). Intuitively, a user choosing to register a security key for login is implicitly requesting resilience to advanced attacks (*e.g.,* real-time phishing). To that user, a service provider should only allow alternatives of equivalent defence capabilities.

Login and recovery are two sides of the same coin. Account recovery techniques (*i.e.,* when a user indicates she is unable to access her account) must not be weaker than login methods. Elevating the allowed authentication alternatives to match the security level of the user-chosen login method must also apply to the configured recovery methods. For a FIDO-enabled account for example, recovery through a secondary email that has weaker security undermines the security of that account. Hammann *et al.* [35] discusses how account-access graphs could help users and service providers discover vulnerable paths.

7.5 User Education

Many participants in our study relied on wrong phishing indicators. Several reported that once they click a link in an email, they wait to see if the visited page is rendered correctly; if not, they become alerted of a possible attack. Participant P18-E said: *"I decide before whether to click or not, and once I click it, it's opened (done)"*. When asked if she continues checking the visited website, she added: *"Not really"*. When asked about detection strategies, participant P9-E said, *"[...] if the website looks fine, I mean the front page, I am not suspicious"*. Similarly, P20-E classified the phishing website as legitimate: *"It's the same because it looks the same up here [refers to logo section], and I would be trusting it's fine"*.

This is not new. And the fact that phishing and similar social engineering tactics rely on users' lack of understanding or incomplete mental models has been well established in previous literature [6, 19]. So long as authentication methods continue to rely on user actions, user education remains a key countermeasure.

Limitation: the rate of technological advancements, and attack evolution, far exceed the pace of user education. The security of web technologies has generally improved in recent years. HTTPS is now widely deployed, and browser vendors address security vulnerabilities in a timely manner. These advancements have lowered the surface for large-scale attacks, making them significantly more expensive. However, our user study shows that users continue to have erroneous mental models that assume malware is easily executed by malicious websites once visited.[8] It is also worth noting that some participants commented about their understanding of the FIDO security keys based on the information sheet we provided them (adapted from Google's *Security and identity products* pages [31], see Sec. 5.1.1). Promoting security keys as phishing resistant by the industry can contribute to developing wrong mental models for users [62], and can thus have adverse effects as users become less attentive to attacks.

8 Related Work

Phishing is an attack vector that falls in the social engineering category, and has been widely studied in the literature. Phishing techniques are very effective to fool even knowledgeable users, and take over accounts [19, 25, 40–42].

2FA schemes. The industry and the academic community has developed several 2FA schemes [44, 45, 49, 50, 60] to protect users' accounts. However, real-time phishing is still very effective to bypass 2FA and automated tools [32] make such attacks simpler, cheaper, and easy to scale. Previous works [22,55] report that phishing is widely employed and preferred by malicious actors, even at hack-for-hire services [55].

FIDO is based on public key cryptography [17] and its benefits are demonstrated in a company setting [48]. The protocol itself is considered secure and it is promoted by the industry as being foolproof phishing-resistant [31]. The research community so far is focused on the usability aspects of FIDO [20, 23, 28, 65] but have not questioned its security in real-world deployments. However, the necessity for alternative 2FA is already emphasized on previous studies [20, 65] because users cannot always complete the FIDO step. On the users side, the possibility of being locked out is reported as

[8] News about 0-day exploits attract a lot of media attention, which possibly sends users exaggerated signals of the popularity of such attacks.

the main obstacle for using FIDO in daily routine [23, 28].

Anti-Phishing ecosystem. Service providers, browser vendors, and other entities have developed an ecosystem to detect and prevent phishing, however adversaries adapt their tools continuously and evade such systems [56–58, 81]. Oest *et al.* [59] report that a phishing campaign is detected nine hours after the first victim, hence spear-phishings that target individuals are much more difficult to be prevented by the ecosystem.

Another line of work [1, 15, 66, 82] focuses on visual similarities between the forged website and the target one, while other [24, 37, 51] try to detect malicious websites based on the URL analysis. Email providers have developed frameworks to filter out phishing emails [21, 36], however attackers still find their way to their target's inbox [55].

Server-side defenses. Online services implement additional systems (R BA engines) on server side to detect phishing attacks and forbid account takeover [20, 75, 76]. However, Campobasso *et al.* [13] present an investigation of a real world deployment of a tool used to take over accounts without being flagged by RBA engines. To limit the consequences of password reuse [10, 18, 26], works [71, 74] have proposed frameworks that allow servers to learn when a password is compromised, while [47] shows that secure implementation of critical protocols, such as TLS is not trivial for developers.

Client side. Password managers are a possible countermeasure to phishing attacks because the credentials are revealed only if the user visits the correct domain. Blanchou and Youn [9] were among the first to report vulnerabilities in password managers. Others [34, 67] describe the challenges of designing and implementing secure extensions, while [79] reported that spoofing the sidebar is effective in phishing the master password as well. Yang *et al.* [80] measured the effectiveness of browser indicators, while [63] show that users lose the ability to detect phishing some period after training.

9 Concluding Remarks

OTP-based 2FA schemes are now amongst the most common to defend against phishing attacks. Being replayable [3], they fail to defend against real-time phishing, where the adversary relays user-submitted OTPs to the legitimate site in real-time. The FIDO alliance has designed challenge-response mechanisms with browser involvement, which enables the inclusion of a website's URL in the challenge. Relaying the response becomes useless, and real-time phishing is thus defeated. U2F is one such standard, where the response is computed on a hardware token. To handle token loss/malfunction, websites commonly allow/force users to register alternative 2FA mechanisms alongside FIDO's U2F. All FIDO-supporting websites in Alexa's top 100 adopt the practice. We ran a user study to test whether a phishing attack that downgrades FIDO to weaker alternatives is effective. Although the study tested U2F tokens, findings (particularly regarding downgrade effectiveness) can extend to other relevant FIDO specifications. We make the following four remarks.

User studies that evaluate attacks must be gracefully executed. Evaluating attacks through user studies is challenging. Participants may fall for said attacks during the study, not because of successful deception, but rather due to participants' lack of investment in protecting assets or misinterpretation of study requirements. If participants' actions were the sole metric, we would have misidentified 88% (instead of 55%) of participants as susceptible to our attacks (Table 2). Such studies should be followed by semi-structured interviews that delicately gauge explanations of participants' actions, without calling attention to said attacks. Results must be compiled within the context of actions and explanations, combined.

Even with FIDO, users remain susceptible to real-time phishing that downgrades FIDO to weaker alternatives. Most participants failed to detect our phishing attacks. Those who succeeded (10%) have done so without the help of FIDO. We found no case by which a participant was close to fall for real-time phishing, but FIDO protected them. Our social engineering involved displaying the FIDO-prompt to the user (the result of which is discarded), followed by a prompt for another 2FA alternative (the result of which would be relayed to the legitimate server in practice). This amassed to what appeared to participants as a three-factor login, which gave an increased sense of (false) security rather than arousing suspicion. The effect of such attacks in practice is exacerbated by two points: (1) users can become less careful seeing more factors, and (2) reassuring wording on login pages (*e.g.,* Google's statement on 2FA pages *"This extra step shows that it's really you trying to sign in"*).

Despite understanding how to use FIDO [28], users do not understand how FIDO protects them. While discussing how they detected our attacks, no participant mentioned relying on FIDO. FIDO protects users when login is granted after using *only* FIDO, not after using FIDO plus other factors. The former prevents real-time phishing and downgrade attacks, the latter enables them. As it is counter-intuitive, no participant appears to have assimilated this concept.

Enabling only FIDO alternatives to FIDO appears to be an effective countermeasure. To address the necessity of allowing alternatives to FIDO's U2F, without enabling downgrade attacks, websites should only allow alternatives of comparable security. Many of the other countermeasures we explored would either expose users to lockouts due to token losses, or continue to make users potentially susceptible to other social engineering variations. Relevant FIDO specifications that are also resilient to real-time phishing (*e.g.,* CTAP2) appear to be suitable alternatives from a security perspective.

Acknowledgments

We thank Sebastian Navas Chaparro for his help with running the user studies.

References

[1] S. Abdelnabi, K. Krombholz, and M. Fritz. VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. In *ACM Conference on Computer and Communications Security (CCS)*, 2020.

[2] A. Abdou, A. Matrawy, and PC van Oorschot. CPV: Delay-based location verification for the Internet. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 14(2):130–144, 2017.

[3] F. Alaca, A. Abdou, and PC van Oorschot. Comparative Analysis and Framework Evaluating Mimicry-Resistant and Invisible Web Authentication Schemes. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2019.

[4] FIDO Alliance. FIDO2: WebAuthn & CTAP. https://fidoalliance.org/fido2/. [Accessed Oct-2020].

[5] FIDO Alliance. Specifications overview. https://fidoalliance.org/specifications/. [Accessed Oct-2020].

[6] M. Alsharnouby, F. Alaca, and S. Chiasson. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, 82:69 – 82, 2015.

[7] M. AlZomai, B. AlFayyadh, A. Josang, and A. McCullagh. An Experimental Investigation of the Usability of Transaction Authorizationin Online Bank Security Systems. In *Australasian Conference on Information Security*, 2008.

[8] D. Balfanz, A. Czeskis, J. Hodges, JC. Jones, MB. Jones, A. Kumar, A. Liao, R. Lindemann, and E. Lundberg. Web Authentication: An API for accessing Public Key Credentials Level 1. https://www.w3.org/TR/webauthn. [Accessed Oct-2020].

[9] M. Blanchou and P. Youn. Browser extension password managers. https://isecpartners.github.io/whitepapers/passwords/2013/11/05/Browser-Extension-Password-Managers.html. [Accessed Aug-2020].

[10] J. Blocki, B. Harsha, and S. Zhou. On the economics of offline password cracking. In *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[11] 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 *ACM World Wide Web (WWW)*, 2015.

[12] J. Bonneau, C. Herley, PC van Oorschot, and F. Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *IEEE Symposium on Security and Privacy (S&P)*, 2012.

[13] M. Campobasso and L. Allodi. Impersonation-as-a-Service: Characterizing the Emerging Criminal Infrastructure for User Impersonation at Scale. In *ACM Conference on Computer and Communications Security (CCS)*, 2020.

[14] K. Charmaz. *Constructing grounded theory*. SAGE, 2014.

[15] T. Chen, T. Stepan, S. Dick, and J. Miller. An anti-phishing system employing diffused information. *ACM Transactions on Information and System Security (TIFS)*, 16(4), 2014.

[16] S. Ciolino, S. Parkin, and P. Dunphy. Of Two Minds about Two-Factor: Understanding Everyday FIDO U2F Usability through Device Comparison and Experience Sampling. In *USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2019.

[17] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz. Strengthening user authentication through opportunistic cryptographic identity assertions. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[18] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The tangled web of password reuse. In *Network and Distributed System Security Symposium (NDSS)*, 2014.

[19] R. Dhamija, JD. Tygar, and M. Hearst. Why Phishing Works. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2006.

[20] P. Doerfler, M. Marincenko, J. Ranieri, Y. Jiang, A. Moscicki, D. McCoy, and K. Thomas. Evaluating Login Challenges as a Defense Against Account Takeover. In *ACM World Wide Web (WWW)*, 2019.

[21] S. Duman, K. Kalkan-Cakmakci, M. Egele, W. Robertson, and E. Kirda. EmailProfiler: Spearphishing Filtering with Header and Stylometric Features of Emails. In *IEEE Annual Computer Software and Applications Conference (COMPSAC)*, 2016.

[22] J. Esparza. Understanding the credential theft lifecycle. *Computer Fraud and Security*, 2019(2):6 – 9, 2019.

[23] F. Farke, L. Lorenz, T. Schnitzler, P. Markert, and M. Dürmuth. "You still use the password after all" – Exploring FIDO2 Security Keys in a Small Company. In *USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2020.

[24] MN. Feroz and S. Mengel. Phishing URL Detection Using URL Ranking. In *IEEE International Congress on Big Data*, 2015.

[25] P. Finn and M. Jakobsson. Designing and Conducting Phishing Experiments. In *IEEE Technology and Society Magazine, Special Issue on Usability and Security*, 2007.

[26] X. Gao, Y. Yang, C. Liu, C. Mitropoulos, J. Lindqvist, and A. Oulasvirta. Forgetting of Passwords: Ecological Theory and Data. In *USENIX Security Symposium*, 2018.

[27] N. Gelernter, S. Kalma, B. Magnezi, and H. Porcilan. The password reset MitM attack. In *IEEE Symposium on Security and Privacy (S&P)*, 2017.

[28] S. Ghorbani Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel. Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication. In *IEEE Symposium on Security and Privacy (S&P)*, 2020.

[29] H. Gomi, B. Leddy, and D. Saxe. Recommended Account Recovery Practices for FIDO Relying Parties. *FIDO Alliance*, 2019.

[30] Google. Google's strongest security helps keep your private information safe. https://landing.google.com/advancedprotection/. [Accessed Oct-2020].

[31] Google. Titan security key. help prevent account takeovers from phishing attacks. https://cloud.google.com/titan-security-key/. [Accessed Oct-2020].

[32] K. Gretzky. Standalone man-in-the-middle attack framework used for phishing login credentials along with session cookies, allowing for the bypass of 2-factor authentication. https://github.com/kgretzky/evilginx2. [Accessed Oct-2020].

[33] E. Grosse and M. Upadhyay. Authentication at scale. *IEEE Security Privacy*, 11(1):15–22, 2013.

[34] JA. Halderman, B. Waters, and EW. Felten. A Convenient Method for Securely Managing Passwords. In *ACM World Wide Web (WWW)*, 2005.

[35] S. Hammann, S. Radomirovic, R. Sasse, and D. Basin. User Account Access Graphs. In *ACM Conference on Computer and Communications Security (CCS)*, 2019.

[36] Y. Han and Y. Shen. Accurate Spear Phishing Campaign Attribution and Early Detection. In *ACM Symposium on Applied Computing (SAC)*, 2016.

[37] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster. PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[38] United States Federal Trade Commission-Consumer Information. How to Recognize and Avoid Phishing Scams. https://www.consumer.ftc.gov/articles/how-recognize-and-avoid-phishing-scams. [Accessed Oct-2020].

[39] INTERPOL. INTERPOL report shows alarming rate of cyberattacks during COVID-19. https://www.interpol.int/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19. [Accessed Oct-2020].

[40] C. Jackson, DR. Simon, DS. Tan, and A. Barth. An Evaluation of Extended Validation and Picture-in-Picture Phishing Attacks. In *Financial Cryptography and Data Security (FC)*. Springer, 2007.

[41] TN. Jagatic, NA. Johnson, M. Jakobsson, and F. Menczer. Social Phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[42] M. Jakobsson and J. Ratkiewicz. Designing Ethical Phishing Experiments: A study of (ROT13) rOnl query features. In *ACM World Wide Web (WWW)*, 2006.

[43] EE. Jones. Content analysis for the social sciences and humanities. *PsycCRITIQUES*, 14(11):615–616, 1969.

[44] N. Karapanos, C. Marforio, C. Soriente, and S. Čapkun. Sound-Proof: Usable Two-Factor Authentication Based on Ambient Sound. In *USENIX Security Symposium*, August 2015.

[45] D. Kogan, N. Manohar, and D. Boneh. T/Key: Second-Factor Authentication From Secure Hash Chains. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[46] K. Krombholz, K. Busse, K. Pfeffer, M. Smith, and E. von Zezschwitz. If HTTPS Were Secure, I Wouldn't Need 2FA - End User and Administrator Mental Models of HTTPS. In *IEEE Symposium on Security and Privacy (S&P)*, 2019.

[47] K. Krombholz, W. Mayer, M. Schmiedecker, and E. Weippl. "I Have No Idea What I'm Doing" - On the Usability of Deploying HTTPS. In *USENIX Security Symposium*, 2017.

[48] J. Lang, A. Czeskis, D. Balfanz, and M. Schilder. Security Keys: Practical Cryptographic Second Factors for

the Modern Web. In *Financial Cryptography and Data Security (FC)*. Springer, 2016.

[49] Google LLC. Google authenticator. https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2. [Accessed Oct-2020].

[50] RSA Security LLC. Rsa securid hard token. https://www.rsa.com/en-us/products/rsa-securid-suite/rsa-securid-access. [Accessed Oct-2020].

[51] J. Ma, LK. Saul, S. Savage, and GM. Voelker. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[52] L. Mathews. Homeland Security Chief Cites Phishing As Top Hacking Threat. https://www.forbes.com/sites/leemathews/2016/11/29/homeland-security-says-phishing-biggest-hacking-threat/#111b1f771978. [Accessed Oct-2020].

[53] M. Maxim and A. Cser. Best practices: Selecting, deploying, and managing enterprise password managers. https://www.keepersecurity.com/assets/pdf/Keeper-White-Paper-Forrester-Report.pdf. Forrester Research. [Accessed Oct-2020].

[54] Microsoft. Set up a security key as your verification method | azure ad. https://docs.microsoft.com/en-us/azure/active-directory/user-help/security-info-setup-security-key. [Accessed Oct-2020].

[55] A. Mirian, J. DeBlasio, S. Savage, GM. Voelker, and K. Thomas. Hack for Hire: Exploring the Emerging Market for Account Hijacking. In *ACM World Wide Web (WWW)*, 2019.

[56] A. Oest, Y. Safaei, A. Doupé, G. Ahn, B. Wardman, and K. Tyers. PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists. In *IEEE Symposium on Security and Privacy (S&P)*, 2019.

[57] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, and A. Doupé. PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists. In *USENIX Security Symposium*, 2020.

[58] A. Oest, Y. Safei, A. Doupé, G. Ahn, B. Wardman, and G. Warner. Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In *APWG Symposium on Electronic Crime Research (eCrime)*, 2018.

[59] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G. Ahn. Sunrise to Sunset: Analyzing the End-to-end Life Cycle and Effectiveness of Phishing Attacks at Scale. In *USENIX Security Symposium*, 2020.

[60] T. Petsas, G. Tsirantonakis, E. Athanasopoulos, and S. Ioannidis. Two-factor authentication: is the world ready?: quantifying 2FA adoption. In *ACM European Workshop on Systems Security (EuroSec)*, 2015.

[61] E. Redmiles. "Should I Worry?" A Cross-Cultural Examination of Account Security Incident Response. In *IEEE Symposium on Security and Privacy (S&P)*, 2016.

[62] E. Redmiles, N. Warford, A. Koneru, S. Kross, M. Morales, R. Stevens, and M. Mazurek. A Comprehensive Quality Evaluation of Security and Privacy Advice on the Web. In *USENIX Security Symposium*, 2020.

[63] B. Reinheimer, L. Aldag, P. Mayer, M. Mossano, R. Duezguen, B. Lofthouse, T. von Landesberger, and M. Volkamer. An investigation of phishing awareness and education over time: When and how to best remind users. In *USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2020.

[64] Google Transparency Report. HTTPS encryption on the web. https://transparencyreport.google.com/https/certificates. [Accessed Oct-2020].

[65] J. Reynolds, T. Smith, K. Reese, L. Dickinson, S. Ruoti, and K. Seamons. A Tale of Two Studies: The Best and Worst of YubiKey Usability. In *IEEE Symposium on Security and Privacy (S&P)*, 2018.

[66] APE. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi. A layout-similarity-based approach for detecting phishing pages. In *IEEE Security and Privacy in Communications Networks and the Workshops (SecureComm)*, 2007.

[67] B. Ross, C. Jackson, N. Miyake, D. Boneh, and JC. Mitchell. Stronger Password Authentication Using Browser Extensions. In *USENIX Security Symposium*, 2005.

[68] S. Schechter, AJB. Brush, and S. Egelman. It's No Secret. Measuring the Security and Reliability of Authentication via "Secret" Questions. In *IEEE Symposium on Security and Privacy (S&P)*, 2009.

[69] S. Schechter, S. Egelman, and RW. Reeder. It's Not What You Know, but Who You Know: A Social Approach to Last-Resort Authentication. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2009.

[70] K. Thomas, F. Li, A. Zand, J. Barrett, J. Ranieri, L. Invernizzi, Y. Markov, O. Comanescu, V. Eranti, A. Moscicki, D. Margolis, V. Paxson, and E. Bursztein. Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[71] K. Thomas, J. Pullman, K. Yeo, A. Raghunathan, PG Kelley, L. Invernizzi, B. Benko, T. Pietraszek, S. Patel, D. Boneh, and E. Bursztein. Protecting accounts from credential stuffing with password breach alerting. In *USENIX Security Symposium*, 2019.

[72] E. Ulqinaku, D. Lain, and S. Čapkun. 2FA-PP: 2nd Factor Phishing Prevention. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019.

[73] PC van Oorschot. *Computer Security and the Internet: Tools and Jewels*. Springer Nature, 2020.

[74] KC. Wang and MK. Reiter. How to end password reuse on the web. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

[75] S. Wiefling, M. Dürmuth, and L. Lo Iacono. More Than Just Good Passwords? A Study on Usability and Security Perceptions of Risk-based Authentication. In *Annual Computer Security Applications Conference (ACSAC)*, 2020.

[76] S. Wiefling, L. Lo Iacono, and M. Dürmuth. Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild. In *ICT Systems Security and Privacy Protection*. Springer, 2019.

[77] H. Wimberly and L. Liebrock. Using Fingerprint Authentication to Reduce System Security: An Empirical Study. In *IEEE Symposium on Security and Privacy (S&P)*, 2011.

[78] M. Wu, R. Miller, and S. Garfinkel. Do security toolbars actually prevent phishing attacks? In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2006.

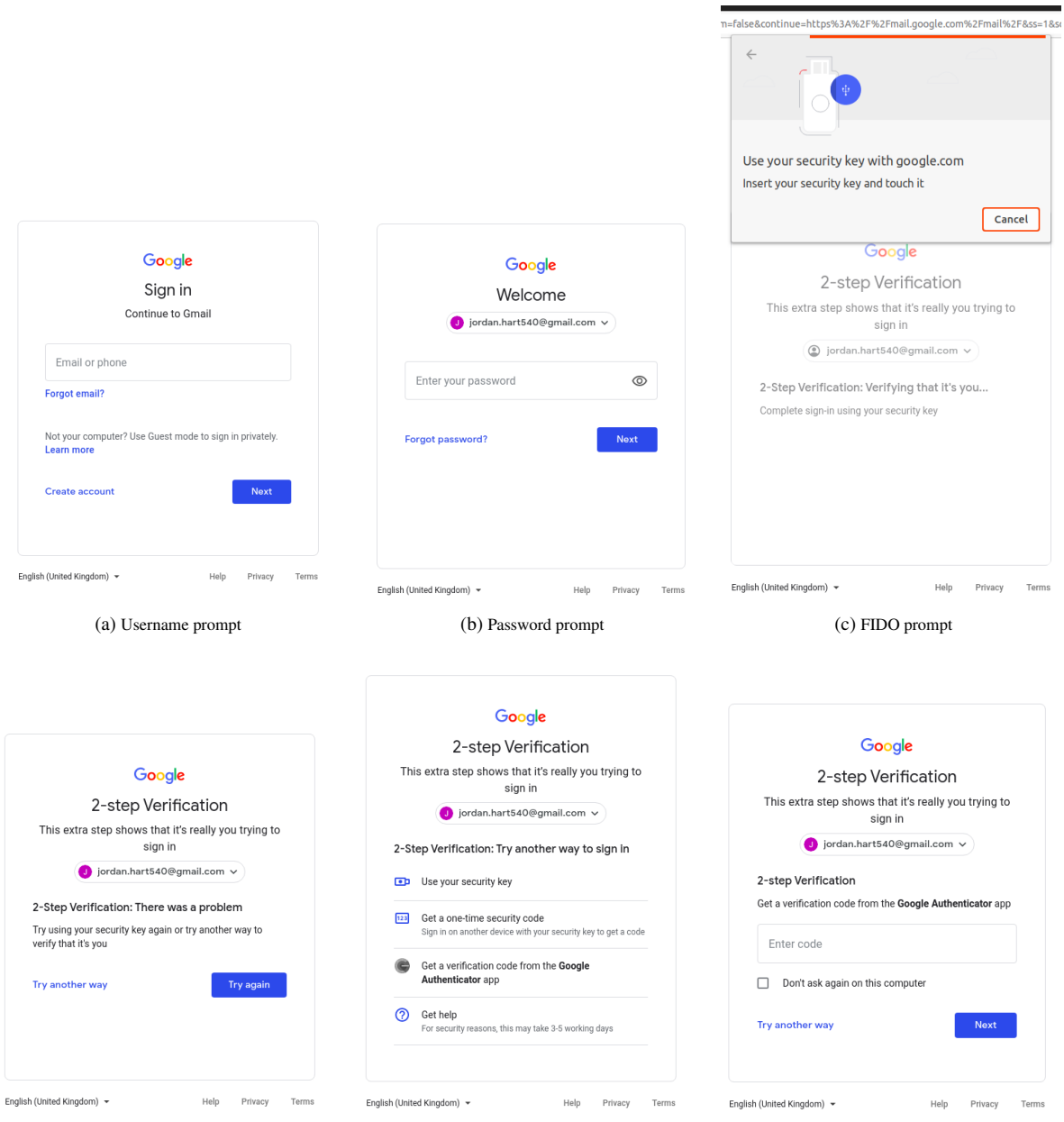
[79] M. Wu, R. Miller, and G. Little. Web wallet: Preventing phishing attacks by revealing user intentions. In *Symposium on Usable Privacy and Security (SOUPS)*, 2006.

[80] W. Yang, A. Xiong, J. Chen, RW. Proctor, and N. Li. Use of Phishing Training to Improve Security Warning Compliance: Evidence from a Field Experiment. In *ACM Hot Topics in Science of Security (HoTSoS): Symposium and Bootcamp*, 2017.

[81] P. Zhang, A. Oest, H. Cho, Z. Sun, RC. Johnson, B. Wardman, S. Sarker, A. Kapravelos, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupé, and G. Ahn. CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. In *IEEE Symposium on Security and Privacy (S&P)*, 2021.

[82] Y. Zhang, JI. Hong, and LF. Cranor. Cantina: A content-based approach to detecting phishing web sites. In *World Wide Web (WWW)*, 2007.

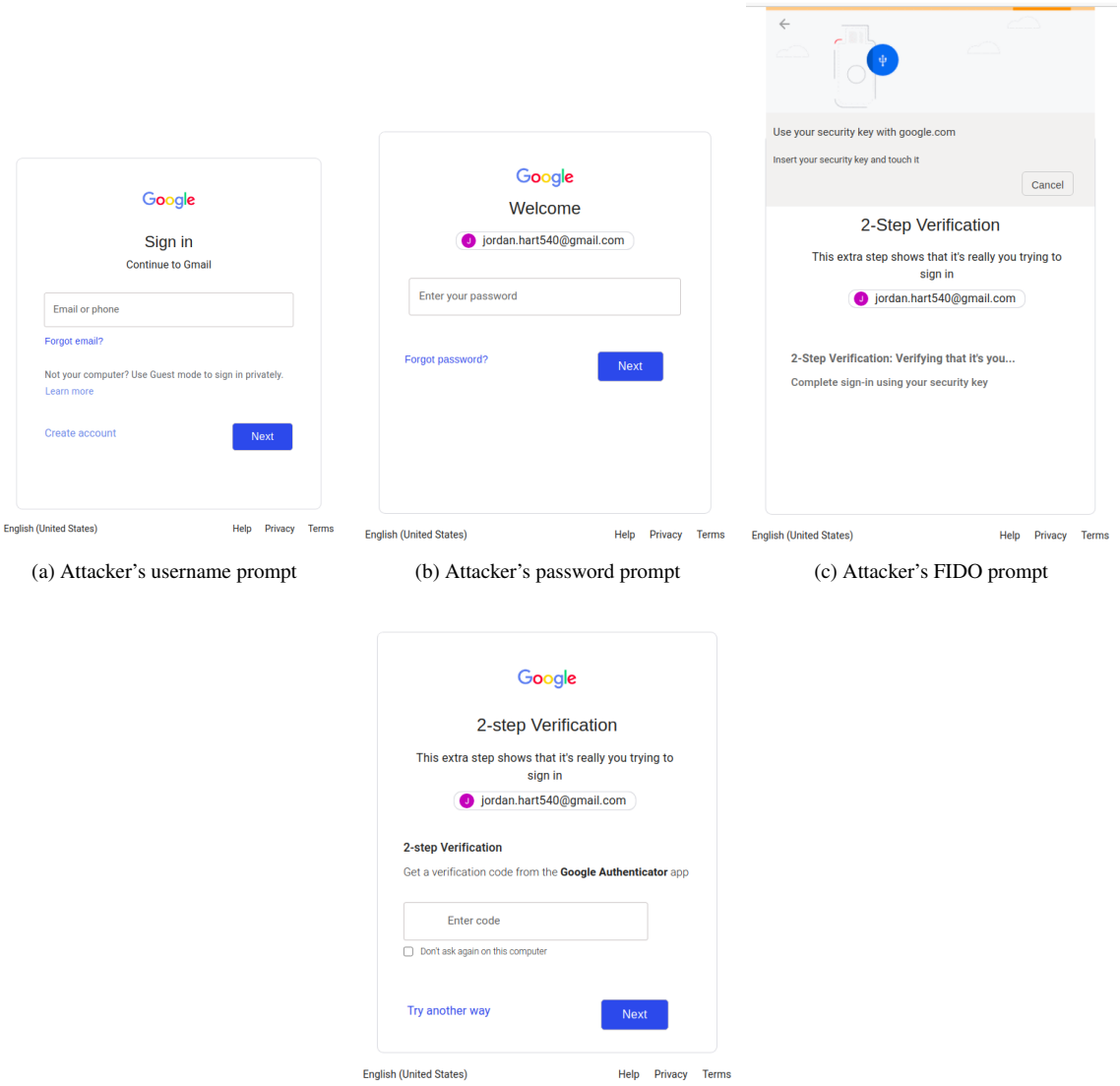
A Webpages



(a) Username prompt (b) Password prompt (c) FIDO prompt

(d) FIDO cancelled (e) Google's alternatives (f) Authenticator prompt

Figure 6: Chronological pages viewed to the attacker while logging-in to Google, upon attempts to impersonate "Jordan Hart", the persona in our user study (Sec.5). Screenshots taken: Aug 20, 2020.



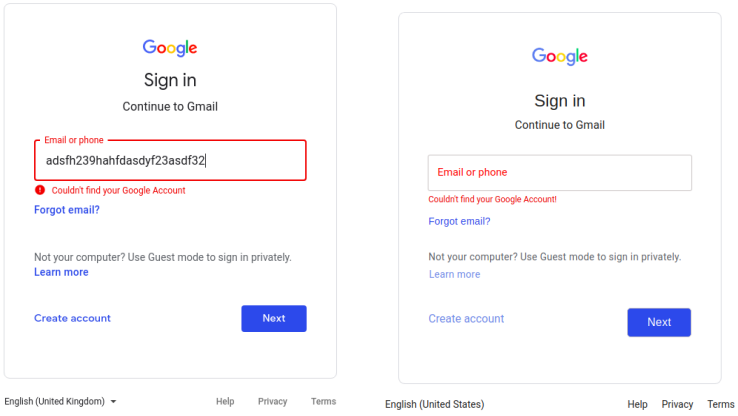
(a) Attacker's username prompt

(b) Attacker's password prompt

(c) Attacker's FIDO prompt

(d) Attacker's Google Authenticator prompt

Figure 7: Attacker's phishing website. Snapshots taken from the website we designed for our user study.



(a) Google's page (b) Our phishing page

Figure 8: Non-existent username.

B Participant Demographics

P-ID	Gender	Age	Highest education level	Occupation	Field of study
P1-N	F	27	Bachelor	Teacher	
P2-N	M	22	Bachelor	Student	Computer Science
P3-N	F	26	High-school	Personal Support Worker	
P4-N	F	22	Bachelor		Law and Legal Studies
P5-N	M	23	Bachelor	Recently graduated	Electrical engineering
P6-N	M	29	Post-grad	Student	Mechanical Engineering
P7-N	M	18	Bachelor	Student	Software Engineering
P8-N	M	19	Bachelor		Computer Science
P9-N	M	29	Post-grad	TA/RA	Psychology
P10-N	F	21	Bachelor	Student	Cognitive Science
P11-N	F	26	Bachelor	Advisor	
P12-N	F	27	Post-grad	Federal government	
P13-N	F	28	Post-grad	UX Designer	
P14-N	M	38	Bachelor	University management	
P15-N	Other	22	Bachelor	Student	Computer Science
P16-N	F	48	Post-grad	Accreditation and QA Officer	
P17-N	M	29	Bachelor	Data Protection	Law and Legal Studies
P18-N	F	28	Post-grad	IT	
P19-N	M	40	Post-grad	PostDoc	
P20-N	M	64	Post-grad	semi retired	
P21-N	M	22	High-school	Student	International Development
P22-N	F	43	Post-grad	International development	
P23-N	F	20	Bachelor	Student	Engineering
P24-N	F	58	Bachelor	Conference coordinator	
P25-N	F	32	Bachelor	Coop Student Advisor	
P26-N	F	26	Bachelor	RA	Wireless Communication
P1-E	F	52	Bachelor	Administrative assistant	
P2-E	F	39	Post-grad	Communications	
P3-E	M		Post-grad	PhD Student	Electrical Engineering
P4-E	F	30	Bachelors	Communications	
P5-E	F	46	Bachelors	Admin Assistant	
P6-E	M	36	Post-grad	Project Leader/R&D	
P7-E	M	22	Post-grad	Student	Computer Science
P8-E	F	26	Post-grad	Finance Associate	
P9-E	M	29	Post-grad	Student	Mechanical Engineering
P10-E	M	40	Bachelors	Information Technology	
P11-E	M	19	High-school	Student	Computer Science
P12-E	F	32	Post-grad	Student	Chemistry
P13-E	F	23	Post-grad	Student	Banking and Finance
P14-E	F	25	Post-grad	Student	Business and Economics
P15-E	F	22	Post-grad	Student	Immunology, Biomedical Sciences
P16-E	M	23	Bachelors	Student	Computational linguistics
P17-E	F	31	High-school	Receptionist	
P18-E	F	19	Bachelors	Student	Environmental Studies
P19-E	M	26	Bachelors	Student	Biology
P20-E	F	27	Post-grad	Student	Social Sciences
P21-E	M	28	Post-grad	Student/Intern	Process Engineering
P22-E	M	25	Post-grad	Student	Pharmacy
P23-E	M	33	Post-grad	Accountant	Economics
P24-E	M	26	Bachelors	Digital Marketing	
P25-E	M	30	Post-grad	Student	Computer Security

C Employee On-boarding Information

NanoTech IT department

Employee Onboarding Information

Welcome to **NanoTech**! We're excited to have you!

At **NanoTech**, we're committed to protecting the company, employees, and customers' resources, internal and external networks, and sensitive data.

Our goals:

- Safeguard **NanoTech** confidential information, employee information, and our customers' confidential information
- Ensure uninterrupted and efficient operations at **NanoTech**
- Protecting **NanoTech** against scammers, including phishing attacks
- Comply with industry, regulatory, and customer requirements

Your role:

- Report theft, loss, or unauthorized disclosure of **NanoTech** information
- Report attempts for stealing **NanoTech** information, including suspicious phishing emails
- Adhere to copyright, trade secret, patent and IP laws
- Log off from your email account(s) at the end of your work day

As part of your onboarding process, you'll receive your work devices and credentials. Reach out to the IT department if you have any issues.

You will be using two-factor authentication to authenticate network resources, computer resources, and Google services. For authentication, you will need your password and your security key. If you lose your security key or if it was damaged, you can login to your account using other backup mechanisms, eg, using the Google Authenticator app already installed on your work phone. In case of loss or damage to the security key, please reach out immediately to the IT department to replace your key.

Why the FIDO security keys?

A FIDO security key is a phishing-resistant two-factor authentication (2FA) device. FIDO keys use cryptography to provide two-way verification: it makes sure that you are logging into the service you originally registered the security key with, and the service verifies that it's the correct security key as well. This provides superior protection to code-based verification, like SMS and one-time password (OTP).

We've already registered your security key with your accounts! Start inventing!

Please sign here to indicate that you have read this information and agree to adhere to it.

***Signature*/Date**

Jordan Hart

Email: jordan.hart540@hotmail.com
Password: [redacted]
(Email is the primary method of correspondence in the company)

Google services account
Username: jordan.hart540@gmail.com
Password: [redacted]
(You will need to use two-factor authentication for Google services)

Manager: Alex James (alex.james1231@hotmail.com)
IT manager: Sam Parker (sam.parker000@hotmail.com)
HR correspondence: Sam Logan (sam.logan2019@hotmail.com)