JOURNAL OF CYBERSECURITY

Journal of Cybersecurity, 2020, 1–15
doi: 10.1093/cybsec/tyaa010
Research Paper

Research Paper

Deploying authentication in the wild: towards greater ecological validity in security usability studies

Seb Aebischer,[1] Claudio Dettoni,[1] Graeme Jenkinson,[1] Kat Krol,[1] David Llewellyn-Jones,[1] Toshiyuki Masui[2,3] and Frank Stajano[1,*,†]

[1]Computer Laboratory, University of Cambridge, Cambridge, UK, [2]Faculty of Science and Technology, Keio University Shonan Fujisawa Campus, Fujisawa, Kanagawa, Japan and [3]Nota Inc., Japan

*Correspondence address: Department of Computer Science and Technology, University of Cambridge, 15 JJ Thomson Avenue, Cambridge, CB3 0FD, UK. Tel: +44-1223-763-500; E-mail: frank.stajano@cl.cam.ac.uk

†Authors listed in alphabetical order. All authors contributed in some form to the Gyazo study while only authors S.A., C.D., K.K., D.L.-J. and F.S. contributed to the Innovate UK study. Author G.J. designed and coded the reverse proxy used in the Gyazo study but left the Pico project before the study started. Author T.M., inventor and CTO of Gyazo, visited the other authors in Cambridge several times for a total of 6 months in connection with the Gyazo study. Authors S.A., C.D., K.K. and D.L.-J. were at Cambridge while they carried out the research described in this article. The Principal Investigator was F.S.

Received 8 March 2019; revised 24 January 2020; accepted 19 May 2020

Abstract

Pico is a token-based login method that claims to be simultaneously more usable and more secure than passwords. It does not ask users to remember any secrets, nor to type one-time passwords. We evaluate Pico's claim with two deployments and user studies, one on a web-based service and another within an organization. Our main aim is to collect actionable intelligence on how to improve the usability and deployability of Pico. In our first study we team up with an established website, Gyazo, to offer this alternative login mechanism to users intent on performing a real task of image sharing. From the lessons of this first study, we retarget Pico's focus from replacing web passwords to replacing desktop login passwords; and thus in our second study we engage with a government organization, Innovate UK, to offer employees the ability to lock and unlock their computer automatically based on proximity. We focus particularly on the ecological validity of the trials and we thereby gain valuable insights into the viability of Pico, not only through the actual responses from the participants but also through the many practical challenges we had to face and overcome. Reflecting on the bigger picture, from our experience we believe the security usability community would greatly benefit from pushing towards greater ecological validity in published work, despite the considerable difficulties and costs involved.

Key words authentication; ecological validity; passwords

© The Author(s) 2020. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

Introduction

It is astonishing that the number of passwords that each of us must use continues to increase year on year, despite users regularly rating passwords as a major inconvenience and security professionals citing weak or reused passwords as some of the most commonly exploited security vulnerabilities. Many talented researchers have proposed innovative alternatives, but any claims of improvement over the usability of passwords must necessarily be validated by user studies. We might wonder why the positive feedback invariably gathered by the creators of new authentication schemes in such studies does not translate into widespread adoption of such schemes and the demise of passwords. A large part of the answer is due to reluctance to change and to the asymmetric incentives for the parties involved: the pain of passwords is experienced by the prover but not by the verifier, for whom passwords are often the cheapest and most convenient option. But another part of the answer might also be that several user studies are insufficiently realistic, and thus excessively optimistic in predicting the success of the new scheme they evaluated.

The key message of this article is that, to assess the usability of a security mechanism (or, more specifically, of an authentication system), there is ultimately no substitute for building it, deploying it and trying it out 'in the wild', with people who will use it while going about their usual daily tasks. This is because, except perhaps for some system administrators, security is rarely anyone's primary goal: the primary goal is to access some resources and get one's job done, whereas the security mechanism, for the user, far from being the goal, is an externally-imposed annoyance that gets in the way of reaching the primary goal. Therefore, studying the usability of the security mechanism in isolation, or with test subjects focused on the security mechanism rather than on their primary task, may hide crucial aspects of the whole story. It might for example hide the fact that users are logging in less frequently in order to avoid the inconvenience of interacting with the authentication scheme. Lab-based and even MTurk-based experiments have their place in the earlier stages of assessing new ideas and designs; but we advocate that, to develop easier to use security systems, we must at some point build and deploy those systems and verify how effectively they actually solve the problems that real people face.

Our team had created Pico (Section 'Background: the Pico user authentication system'), an authentication solution designed to be more secure and usable than passwords. In this article, we report on two real-world deployments of Pico, intended to assess our own claims about the system and, more generally, to improve Pico's fitness for purpose. The first deployment (Section 'The website authentication study with Gyazo') was run in collaboration with Alexa Top 500 website gyazo.com: Gyazo customers who participated in our trial would log in to their account by scanning a QR code with their smartphone rather than entering their username and password. In collaboration with the service, we selected a suitable subset of the customer base and ran surveys and interviews with them, before and after monitoring their interactions with the site for two weeks. The issues we encountered during this trial convinced us that the Gyazo website, and perhaps websites in general, were not a setting in which users felt the pain of passwords particularly strongly (Section 'Discussion: the pain of passwords on the web') given their available and commonly used mitigations. We had to swallow our pride and admit we had been attempting to solve a problem that users didn't really experience to any great degree. We therefore retargeted our efforts towards a different setting (Section 'The proximity-based laptop-unlocking study with Innovate UK'): employees logging into the laptop provided by their organization, where we expected security policies to be enforced more strictly, password managers not to be available and thus the pain of passwords to be greater. We partnered with Innovate UK, a government agency, pivoting from website login to operating system login and porting our software to new platforms: Windows for the back-end and iOS for the phone client. We also introduced a new, simpler mode of operation: proximity-based continuous authentication. By monitoring the Bluetooth signal from the user's smartphone, we automatically unlocked the computer in the presence of its user and locked it otherwise. This mode seemed particularly well suited to the open plan offices common in corporate environments. We encountered many problems in this second trial, which we document here to help other researchers stay clear of them.

The main contributions of this article are as follows:

- We report on the first user study of Pico with actual users of a popular website (Section 'The website authentication study with Gyazo').
- We highlight the additional insights offered by having adopted a setup with high ecological validity (Section 'Discussion: the pain of passwords on the web'), which partly explain why many password replacement schemes, including ours, get little traction. Some of these explanations may sound obvious in hindsight, but they were not to us at the time. We identify contexts where Pico might do well in comparison to passwords or other alternatives.
- We use the intelligence thus gained to target a new and more appropriate scenario: no longer logging into a web account but automatically unlocking and re-locking a corporate laptop within an open-plan office (Section 'The proximity-based laptop-unlocking study with Innovate UK'). Towards that, we radically revisit the design of Pico and develop new software for it.
- We document our technical solutions, including the reverse-proxy technique we created for allowing a website to offer a Pico login without having to modify the website server itself (Section 'Website login'). We release the Pico software as open source for other researchers to experiment and build upon (Section 'Open source').
- We candidly admit to our frustrations and failures (Section 'The proximity-based laptop-unlocking study with Innovate UK'), alongside reporting on what went well. Our philosophy of building a system, deploying it in a real environment and measuring its effectiveness for real users has a high cost and a lower probability of success, and we do not attempt to hide that. But we still consider it worth the investment, and hope the security usability community will adopt it more frequently in the future.

Background: the Pico user authentication system

Original design

Pico grew out of the basic observation that the password directives commonly given by techies at the time (long, complex, and unguessable passwords; all different; and a prohibition to write them down) were mutually contradictory and thus ultimately *unethical*, in that they blamed the user for not doing something that could not reasonably be done. In its original clean-slate design, first presented by Stajano in 2011 [1], Pico consisted of a single-purpose hardware token that would remember arbitrarily many public key credentials on behalf of its owner, a different one for every account, and would stay locked and refuse to work unless it was within the authentication aura of its owner, as defined by the presence of wireless wearable accessories called Picosiblings.

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

Stajano secured a generous grant from the European Research Council and led an evolving team of researchers who, over five years, incrementally prototyped, built, tested and improved upon various aspects of this design. Practical constraints, experience from previous trials, and sometimes the need to compromise on the original clean-slate design for interoperability with deployed systems, all contributed to a gradual evolution and maturation of Pico's design.

Logging into a website originally involved, on the back-end, augmenting the login page with a QR code that contained the public key of the website. The user wishing to log in would scan that code with their Pico hardware token device (similar in functionality to a smartphone, but single-purpose to minimize the attack surface) and, provided the Pico device recognized the public key as that of a website it had an account on, it would run a mutual authentication protocol [2], based on Krawczyk's well-established Sigma-I [3], in which the Pico device would offer to the verifier its own public key for that account.

We obviously faced the bootstrapping problem that no real website would be running the Pico public key protocol until Pico was already widely deployed, and vice versa. To break the vicious circle we developed a browser plug-in, the Pico Lens [4], that made regular websites appear to the user as if they supported Pico, by displaying a QR code for them on their login page. The Pico client device would then effectively run the authentication protocol with the Pico Lens. In turn, the Pico Lens would authenticate the user to the real website by sending it a stored password, similarly to what a native in-browser password manager does. That way, from a user experience viewpoint, all websites would appear to support Pico, whether they did so natively or not. (Of course those that didn't would not benefit from the additional security of public keys over static passwords.) A limitation of this technique was that it involved fragile heuristics to detect what was a login page and how to type username and password into it. We thus also designed and proposed a structurally better solution that would have also helped other password managers [5]. We also evolved a variant of Pico that no longer required public key cryptography [6].

We explored various ideas around the Pico security tokens [7] and debriefed users on their acceptance of them using non-functional prototypes made of plasticine and Polymorph [8]; in parallel, we developed various functional smartphone-based prototypes of the software in the Pico client and Picosiblings. We eventually abandoned the dedicated single-purpose hardware token for a more mundane smartphone app when it became clear through focus groups and user studies that few regular users would be willing to carry a new gadget, however secure, and that they would naturally prefer to trade security for convenience.[1]

Website login

The version of Pico that was used for our first public deployment (Section 'The website authentication study with Gyazo') consisted, on the client side, of an app for Android smartphones. On the server side, if we were going to earn the collaboration of a real website, we anticipated we would not be allowed to make any changes to their production server. The Pico Lens would have worked, but it would have required users to adopt the specific web browser (Firefox) for which we had developed the plug-in; it would have also forced them always to log in to their web account from the same computer, whereas we would have preferred to allow them to roam between computers and even log in from (say) an Internet café. This is because we expected that, while logging in from the same computer, they probably would not be forced to enter their password very often; either because of a long-lived login cookie or because their browser's password manager would remember the password for them.

We thus developed an alternative architecture in which, instead of putting a Pico proxy in front of the client (the Pico Lens embedded in the user's web browser), we put one in front of the server. In this 'reverse proxy' design the user could log in from any browser, without any plugins: they would just have to visit our custom address https://pico.gyazo.com instead of the regular https://www.gyazo.com. Our proxy web server would display a QR code, perform mutual authentication with the user and then embed an authentication cookie provided by Pico into the browser session, before handing over the authenticated user to the real Gyazo website [9].

We built our reverse proxy on top of NGINX and we injected client-side Javascript using a `sub_filter` rewrite rule. This allowed us to add the Pico QR code onto the Gyazo site's usual login form without altering any of the backend code.

As expected, the fact that we did not require Gyazo to modify their server in any way, and indeed that users not taking part in the trial would not even be aware of the existence of the Pico login option, was a crucial facilitating factor in Gyazo's decision to run the trial with us.

Laptop login

For reasons that will be explained in greater detail in Section 'Discussion: the pain of passwords on the web', our next Pico deployment was no longer to the (external) users of a website but to the (internal) employees of an organization. We would now be logging them into their corporate Windows laptop. This of course involved a total rewrite of the Pico authentication back-end, as well as its integration into Windows. Moreover, a majority of those employees used iOS rather than Android, so the client side had to be ported as well. We thus implemented a version of Pico in which the Windows login screen would display a QR code: acquiring it with the Pico app on your iPhone would then log you in.

In this context of local (rather than across-the-web) login, we ventured into continuous authentication, part of the original design of Pico: the idea of unlocking a device when within the aura of its owner. When the laptop sensed that the user's smartphone was nearby, it would unlock; and when the user's phone went out of range, the laptop would lock. To sense proximity, we relied on Bluetooth Low Energy RSSI (Received Signal Strength Indication). This in turn limited the hardware we could use to models of laptop and phone that implemented the relevant Bluetooth LE specifications. Fortunately, this included the models of Lenovo laptops and Apple iPhones issued to employees at our partner organization Innovate UK. At this stage of exploration of the feasibility and acceptability of various regions of the design space, we deemed usability much more important than security and we therefore did not worry about guarding against relay attacks.

[1] The banking industry drew similar conclusions. Banks are now happy to offer smartphone banking apps to their customers, lest they switch to the competition.

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

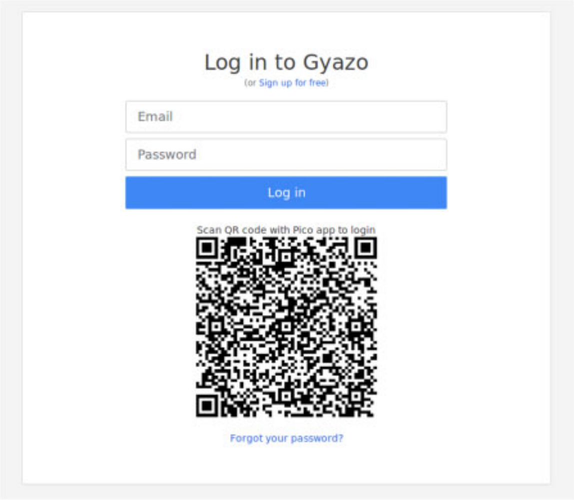


Figure 1: the Gyazo login screen with a Pico QR code

Open source

We documented and released the Pico software as Open Source at https://github.com/mypico, including the various Android, iOS and Windows versions used in the trials described in this article. The software is offered as is, without any maintenance or support since all the team members have now moved on to other projects, but in the hope it will be useful to other researchers. If you make use of the Pico software in your research, please acknowledge it by citing this article.

The website authentication study with Gyazo

Background

About Gyazo

Gyazo,[2] an Alexa Top 500 website, is an image capture and sharing service that offers two primary modes of operation. Users can either capture a portion of their desktop as a screenshot using an application installed on their computer, or upload images captured through other means directly to the site. The website interface then allows images to be organized, tagged and shared with others, but Gyazo has focused particularly on providing a frictionless workflow for capturing partial screenshots. The user simply launches the application, at which point they can drag a bounding box over the screen to capture an image. The interface is minimal: the mouse cursor switches to a crosshair design and once the drag is complete the image is automatically uploaded and displayed in the browser. The authentication process is subtle. The Gyazo application stores a non-expiring identity token that it uses to authenticate when uploading an image. This is stored in the user's home folder so that images can be uploaded without any further credentials being requested from the user. While this token allows images to be uploaded, it does not allow any deeper web access to the user's account. For this, the user must log in manually using username and password. The gyazo.com website also uses browser cookies to maintain a login session; these cookies expire after one month.

Our Pico implementation does not affect image upload and the identity token, but rather focuses on access to the website. As shown in Fig. 1, Pico adds a QR code to the login page alongside the traditional username and password fields. The user loads up the Pico app on their phone and scans the QR code. A few seconds later, they are authenticated and the website automatically refreshes to show that they are logged in. Pico uses the data in the QR code to trigger a background authentication, and from the user's perspective the site refreshes and moves from the login page to their account dashboard.

Our experiment

We conducted a three-part user study exploring the usability, deployability and perceived security of Pico when used for authentication to Gyazo. We targeted the subset of Gyazo users who, based on their usage patterns, would find Pico most useful and asked them to use Pico to log in to Gyazo for a period of two weeks. We collected participant feedback and produced a rich set of quantitative and qualitative data including telemetry, ratings, free-text responses and interviews. The number of people involved at each stage is detailed below and summarized in Table 1. We consistently refer to these N_x throughout the article.

In the first stage, out of the $N_0 > 9$ million active users of Gyazo, we asked the company to tell us which ones logged into the Gyazo website with relatively high frequency (defined as 'manually entered their Gyazo password at least once during the previous week'). To the resulting $N_1 = 1136$ users we sent a questionnaire, whose purpose was to understand why they logged in more frequently than other users; of these, $N_2 = 268$ opened it and $N_3 = 85$ completed it. The demographics of these $N_3 = 85$ are in Table 2.

In the second stage, we narrowed down to those willing to participate in a trial, who owned an Android phone and who matched our demographic criteria for a representative sample of the population. We invited the resulting $N_4 = 29$ users to download the Pico Gyazo app and take part in a trial. Of these, $N_5 = 12$ downloaded the app and $N_6 = 11$ of them completed setup and used it for two weeks, thereby participating in the second stage of the study. All of them, $N_7 = 11$, completed an exit questionnaire. Their demographics are in Table 3.

In the third stage, the $N_7 = 11$ Stage 2 participants were invited to take part in a debriefing interview. $N_8 = 7$ agreed to speak with us and we were eventually able to arrange and conduct interviews with $N_9 = 5$ of them (Table 4).

The two questionnaires and the script for the semi-structured interviews are included in the online Supplementary Data that accompany this article, while the code of the app is on GitHub (Section 'Open source').

Experimental methodology

Apparatus

Participants install our Pico Gyazo app for Android, a 4.16 MB download, via Google Play. The first time users open the app, they are invited to enter their Gyazo username and password (Fig. 2, left), which are remembered by the app for future logins. Users are then presented with a QR code scanner (Fig. 2, right). From this point on, until users clear the app data (which only one of the users did during the trial), opening the app will bring them directly to the scanner page. The interface is simple and consists of just this one screen.

We peppered both the app and the proxied Gyazo login page with keen.io[3] calls to collect event-based telemetry data. These

[2] The Gyazo service is offered at the https://gyazo.com/ website and is maintained by Nota Inc.

[3] https://keen.io/

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

allowed us to measure timings for various stages of the login process, including startup, configuration, scanning the QR code and completion. The full set of events is shown in Tables 5 and 6 for the Pico app and login page respectively.

For the configured phone we were able to form an identifier for the user, generated as a salted hash of their Gyazo username and sent with each event. This ensured that data were kept anonymized as they passed through keen.io (since publicly the hashes cannot be reversed), but could be de-anonymized by us given our knowledge of the participants' Gyazo usernames. However, events generated for the unconfigured app and for the website had no access to the username. Consequently, we also collected the IP address and a random channel identifier freshly generated during every connection handshake between the app and browser. This allowed us to correlate events even where the user was not immediately identifiable.

Data analysis

Owing to the small sample sizes in our study, we report on the quantitative results using descriptive rather than inferential statistics.

Table 1: an overview of the number of people during the different stages of the Gyazo study

Number of people		Description of stage
N_0	>9 million	Users on Gyazo
N_1	1136	Were contacted based on frequency of login
N_2	268	Opened the invitation
N_3	85	**Completed the initial questionnaire** (Stage 1)
N_4	29	Were invited to download the Pico Gyazo app for Android
N_5	12	Downloaded the app
N_6	11	Logged into Gyazo using Pico for two weeks
N_7	**11**	**Completed the exit questionnaire** (Stage 2)
N_8	7	Agreed to be interviewed
N_9	5	**Were interviewed** (Stage 3)

In Stage 3, we conducted $N_9 = 5$ in-depth interviews that were audio recorded and later transcribed to provide qualitative data. The interviews lasted 26 minutes on average (range: 19–36) and the transcripts were on average 3917 words long (range: 3099–5637).

The transcripts were analysed using Braun and Clarke's thematic analysis [10] as follows. Two researchers coded the first interview independently and then created a joint codebook, based on which they coded the remaining four interviews. After this, they merged their codebooks and re-coded all five interviews in line with this codebook. The inter-rater reliability was high with a Cohen's Kappa coefficient of 0.84, which is considered to be excellent in statistical textbooks such as Fleiss *et al.* [11].

Research ethics

The study was conducted after having been approved by the Ethics Committee at the University of Cambridge Computer Laboratory (approval number: 384; approval date: 2016-07-28).

Results

Initial questionnaire

When we refer to participants in the $N_3 = 85$ questionnaire, we speak of 'respondents' abbreviated as 'R01' for 'Respondent 1'.

We asked the respondents how often they manually typed in their Gyazo password: most did so less than once per week (Table 7).

Table 3: demographics (gender, age and location) of the $N_7 = 11$ Stage 2 participants

Male	8	18–24	9	USA	6	Greece	1
Female	1	25–34	2	Japan	1	Latvia	1
Prefer not to say	2			Brazil	1	Spain	1

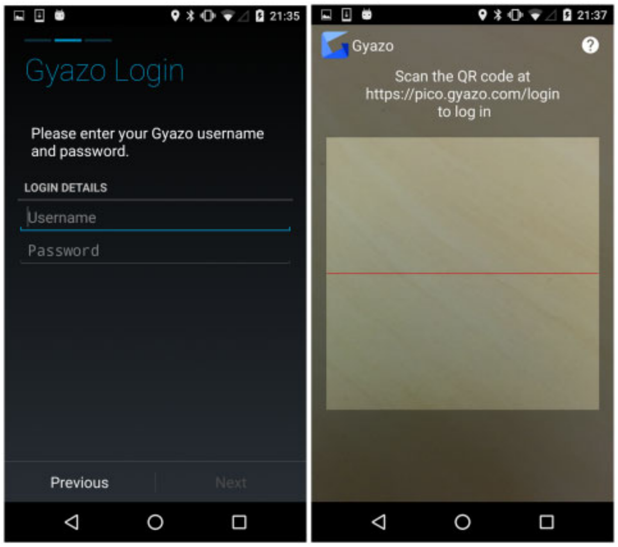
Table 4: demographics (gender, age and location) of the $N_9 = 5$ Stage 3 participants

Male	2	18–24	4	USA	4
Female	1	25–34	1	Latvia	1
Prefer not to say	2				

Table 2: demographics (gender, age and location) of the $N_3 = 85$ Stage 1 participants

				USA	26	Norway	2	Greece	1
				Japan	13	Bangladesh	1	Italy	1
		18–24	56	UK	12	Brazil	1	Latvia	1
Male	69	25–34	18	Canada	7	Colombia	1	Portugal	1
Female	12	35–44	7	Russia	3	Czech Republic	1	Romania	1
Prefer not to say	4	45–54	2	Israel	2	Estonia	1	Spain	1
		55–64	2	Lebanon	2	France	1	Sweden	1
				Netherlands	2	Germany	1	Taiwan	1
								Vietnam	1

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020



Please enter your Gyazo username and password.

Scan the QR code at https://pico.gyazo.com/login to log in

Figure 2: the configuration screen (left) and scanner interface (right)

Table 5: events generated by the Pico app

Starting unconfigured
Starting configured
Configuration complete
Scanned QR code
Scan cancelled by user
Attempting login
Successful authentication
Checking username and password
Credentials checked out
Incorrect credentials
Open credentials page

Table 6: events generated by the login page

Page loaded and updated
Error getting new channel
Pico accepted message on Rendezvous channel
Message from Pico authenticated

Table 7: self-reported frequencies with which respondents manually typed in their passwords on gyazo.com ($N_3 = 85$)

Less than once per week	59	69.4%
At least once per week but less than once per day	8	9.4%
Roughly once per day	9	10.6%
Roughly 2–4 times per day	7	8.2%
Roughly 5 or more times per day	2	2.4%

Next, with an open-ended question, we asked them to explain the main reason why they had to enter their password; 65 of the $N_3 = 85$ provided an answer to this question. Switching to another device was mentioned in 19 cases; for example, R82 wrote: '*Signing in on a different computer*'. Seventeen respondents explained that they needed to enter a password in order to log in and access their images. Our intention had been to understand why respondents had to enter their password given the service sets long-lived cookies: their literal interpretation of the question suggests we should have phrased it more carefully. In 15 cases, respondents stressed they had to enter their password after clearing their cookies. R54 explained: '*Because i clear my cache at the end of the day to speed up the browser*'. Seven respondents mentioned having to log in to Gyazo when switching to another browser. Six respondents said their perception was that they were never asked to enter their password. In four cases, the respondents said that they had to enter it if their password manager/browser failed to do it for them; R57 explained: '*SafeInCloud doesn't actually work*'. Three respondents mentioned they entered their password manually because of security reasons; R17 wrote: '*I never use the option in Chrome save password because of that and also I think its safer to type the password everytime*'. Interestingly, two respondents also mentioned they logged out intentionally in order to practise their password, as R18 explained: '*To better memorize it*'.

Table 8: self-reported methods by which Gyazo study respondents managed their passwords ($N_3 = 85$, multiple choices allowed)

Let my browser remember my password	63	74.1%
Password manager/plugin/extension (e.g. LastPass, 1Password)	19	22.4%
Password generator	13	15.3%
Reset password by email when I need to log in	14	16.5%
Password containing personal information	4	4.7%
Stored in a file/on a piece of paper	10	11.8%
Using the same password on multiple sites	30	35.3%
No special methods (I just remember all of my passwords)	17	20.0%
Other	10	11.8%

Table 9: a break-down of steps needed for an authentication event using the Pico app to log in to Gyazo, with durations in seconds. The duration of each step is the interval between a 'from' and a 'to'

Step no.	Authentication step: from, to	Mean
Step 1	Start Pico app, load page; or: load page, start Pico app	34.55
Step 2	Start Pico app, scan QR code	9.62
Step 3	Scan QR code, successful authentication	2.40
Step 4	Successful authentication, confirmation from website	0.93

We asked respondents what their password management methods were. They could choose from a list of eight options and add their own (Table 8). The most popular password management strategy was letting the browser remember the password, employed by 63 respondents (74.1%), followed by 30 respondents (35.3%) who used the same password across multiple sites and 19 (22.4%) who used a password manager / plugin / extension.

Telemetry results

The Pico app was downloaded by $N_5 = 12$ participants but one participant never used it to authenticate. Overall, we recorded 45 authentication events across $N_6 = 11$ active participants ($M = 4.1$ authentication events per participant; range: 1–14).

For each authentication event, our telemetry collected the timings for the following authentication steps: starting the app, loading the Gyazo login page, scanning the QR code, successful authentication and confirmation from the website (*cfr.* Tables 5 and 6). An authentication event lasted an average of 47.5 s (range: 8–292). Of the 45 authentication events recorded, three were missing telemetry data for some steps and were therefore excluded from subsequent analysis. Out of the remaining 42 fully recorded events, 23 started with the participant opening the Pico app and then opening the page, while 19 started with them loading the

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

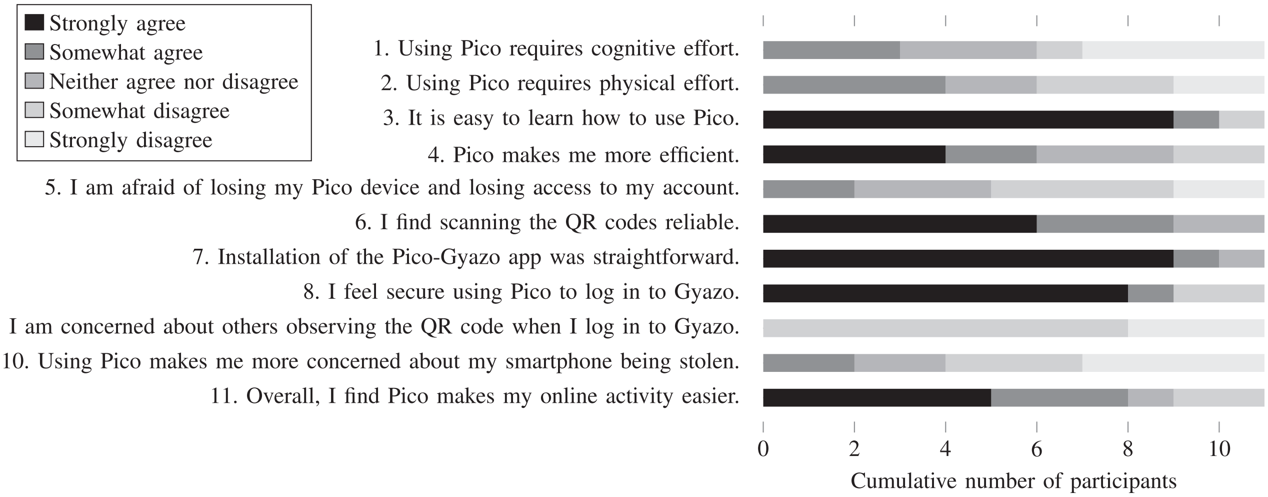


Figure 3: statements capturing aspects of the user experience of Pico in the exit questionnaire of the Gyazo study ($N_7 = 11$). The darker the shading, the more the participant agreed with the statement

page and then opening the app. Table 9 shows the average times for each of the steps. For the first step, if participants started by opening the app and then went on to open the Gyazo website, it took them on average 35.2 s, whereas the other way around it was 33.7. All four steps depended on the response time of various computers and networks, but Steps 1 and 2, which feel quite slow, also depended on the participants' speed. The Pico protocol involves the usual POST request—sent by the Pico to the website—needed to authenticate the user, followed by four messages sent between the Pico and the user's browser (two in each direction) to securely install the cookie. The timings of Steps 3 and 4 are in line with what we would expect to see.

Questionnaire and interview results

Since both questionnaires and interviews touched upon similar themes, we group their results by theme. As above we refer to participants in the questionnaires ($N_3 = 85$ or $N_7 = 11$) as 'respondents' abbreviated as 'R01'. When we refer to participants in the interviews ($N_9 = 5$), we speak of 'interviewees', abbreviated as 'I01'.

Primary task—Gyazo

We began each interview by asking the interviewee about their use of Gyazo, in order to ground their perceptions of Pico in the primary task of using the service. Two interviewees had been using the service for two years, two for one year and one only signed up a month before the study. All interviewees used it for personal use, and one, an artist, also used it for professional reasons. Four interviewees stated that they were not currently premium users; one told us that they used to be a premium user but no longer found it affordable. Two mentioned that they used Gyazo on a daily basis.

We asked the interviewees in what situations they had to enter their password. Two stated that they had to enter it when they deleted cookies; two said that they needed to enter it when they switched devices or shared them. Most people stay logged in long-term, and as a result three interviewees reported that they had to log out deliberately to use Pico in the trial.

Perceptions of Pico

Three interviewees found Pico to be easy to use. Two described it as 'fairly quick', and two as convenient. I04 explained: '*I thought it was very effective, it was very quick, very easy, convenient… I definitely, I like the idea versus having to put in the password every time*'. I02 explained how the swift login with Pico was an encouragement to log in more often: '*I used [Pico] on my college computer a couple of times, and it was just more convenient […] like I need to show a fellow student an image or something like that, so it was just a lot easier to just pop onto the desktop and scan in, so, slightly more, yeah, than using a password*'. Two interviewees also described Pico as 'cool'.

In the exit questionnaire (Stage 2), we asked the respondents to what extent they agreed with eleven statements about Pico on a five-point Likert scale from 'strongly agree' to 'strongly disagree' (Fig. 3).

Respondents disagreed the strongest with statement 9, expressing they were not concerned about others observing the QR code when they logged in to Gyazo. Respondents agreed the strongest with statement 3, saying it was easy to learn how to use Pico and statement 7, that the Pico-Gyazo app was straightforward to install.

The scores for cognitive effort tended to be low (meaning good), with four participants indicating that they disagreed strongly with statement 1 (i.e. they felt Pico does not require cognitive effort). In the interview, we asked one participant who agreed with the statement (score: 2) to elaborate on their rating. It turned out that the participant was unsure about the meaning of the word 'cognitive'. After an explanation of what it meant, they revised their score saying: '*I don't think there's any [cognitive effort], because you don't really have to remember anything, you just have to unlock, you know, your password on your phone, and it's pretty much you use it daily, so the chances of you forgetting it is pretty much non-existent*' (I03).

Familiarity with QR codes

Three participants said that they were already familiar with scanning QR codes: I01 had used them with the Nintendo 3DS, I05 as part of the LINE messaging app, and I02 as part of a school project. Two interviewees found QR codes inconvenient because of the fact that scanning them required having their phone at hand. I01 explained: '*I worked with QR codes before so it wasn't too hard to work with. It just seemed a little bit inconvenient. I mean, getting out my phone and, 'cos I don't usually have my phone on me when I'm at the computer, it's usually somewhere else, so I had to bring*

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

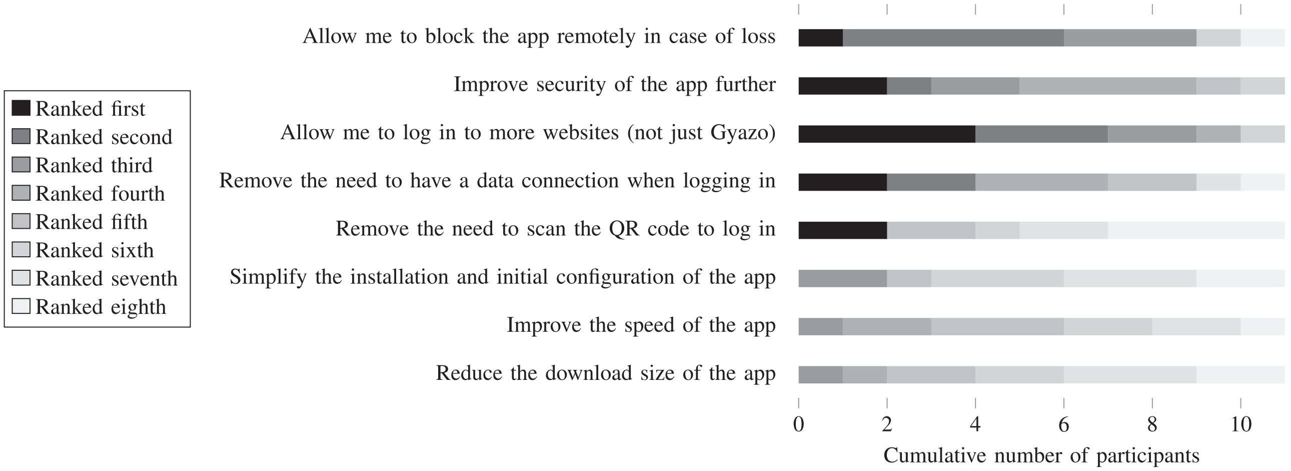


Figure 4: Gyazo study participants' ranking of the different improvements that could be made to Pico ($N_7 = 11$). The darker the shading, the higher the priority ranking of the improvement

my phone over and I had to scan the screen'. I03 experienced some problems with scanning a QR code because of a low quality phone camera and monitor, saying: '*the problems were coming I think from my monitor, my old one, I had the really old one, the big one, CRT monitor, which basically, every time I tried to scan it, it was flickering, so it'd make it harder to scan it*'.

Pico *versus* passwords

We asked our interviewees to compare Pico to passwords. Two participants found Pico was more convenient, and two found it was faster than passwords. Three interviewees thought that Pico was more secure than passwords. When asked about the security of Pico, I04 said: '*I think I put it as around the same*' but then explained that in its current state, Pico might be more secure because it had not been a target for attackers yet: '*I don't really know what the exact basics are between getting into an account using someone's password, so not exactly sure how someone would hack you using Pico, but I guess technically it would be more safe because the technology isn't out yet, but they'd figure it out eventually, if it becomes a major thing*'.

Two interviewees argued that passwords were better because Pico was slower than entering a password. I03 thought that passwords were more secure because they existed only in the user's memory: '*if you have a secure password and you're pretty much the only one person who knows it, I think that's the most secure thing you could possibly have*'.

Suggestions for improvements

In the exit questionnaire, we provided respondents with a list of eight possible improvements that could be made to Pico and asked them to rank these in order of priority from the most to the least important (Fig. 4). The improvement ranked first most often referred to introducing login with Pico to more websites than just Gyazo; it was ranked either first or second by seven out of eleven respondents. The issue that was ranked bottom most often (four times) was removing the need to scan the QR code, although it was also ranked first by two participants.

Apart from the ranking task, respondents were also asked if they had any other suggestions. Four respondents made suggestions related to security, specifically login security and recovery from loss. The login-related suggestions included requiring the user to enter a username as an extra hindrance to potential attackers and requiring two-factor authentication in certain contexts. The suggestions relating to recovery from loss included creating a '*button to disable the Pico app in the website if the phone is lost or stolen*' (R01) and having a fall-back login method if the phone is lost. Otherwise, two respondents stressed that they would like to see Pico integrated with more websites. R10 explained: '*Just more websites, this is way more easy to login into websites*'. One respondent suggested other modes of logging in apart from scanning a QR code, such as '*timing tap and voice recognition*' (R03).

Table 10: preferred ways to lock phone, with number of Gyazo study participants who chose it ($N_7 = 11$, multiple choices allowed)

4-digit PIN	3
6-digit PIN	5
Password	3
Pattern lock	6
Fingerprint	6
Slide lock	1

Phone use habits

We also asked participants about their phone use habits to gauge how Pico fitted into their routine. Two interviewees reported using a pattern lock, two using a 4-digit PIN, and one not having a lock at all. Participants provided all kinds of reasons for preferring one method over another. I02 explained: '*I don't particularly care for pattern locks because I have rheumatoid arthritis. So, using my thumbs in that particular way is a bit painful, so for convenience like I don't...*' When asked what they were using instead they told us: '*I have a PIN on it. [...] But, but it's also the PIN to my debit card!*' The interviewee then went on to discuss how they thought fingerprint scanners were the most convenient way of logging in. When asked to compare the security of the different methods of locking their phone, I02 responded: '*I know that they can be compromised, but at the same time I don't necessarily think that affects most users. Most people aren't going to be maliciously attacked and have their fingerprint stolen, but I think in higher security situations that might be a problem. I wouldn't, if I were say a diplomat, or something like that, I don't think I would trust my fingerprint that much you know, but if it's just an average Joe, sure, why not.*'

In the exit questionnaire, we asked the respondents what method of locking they would use if all their passwords were stored on their phone. We provided them with six possibilities, of which the most

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

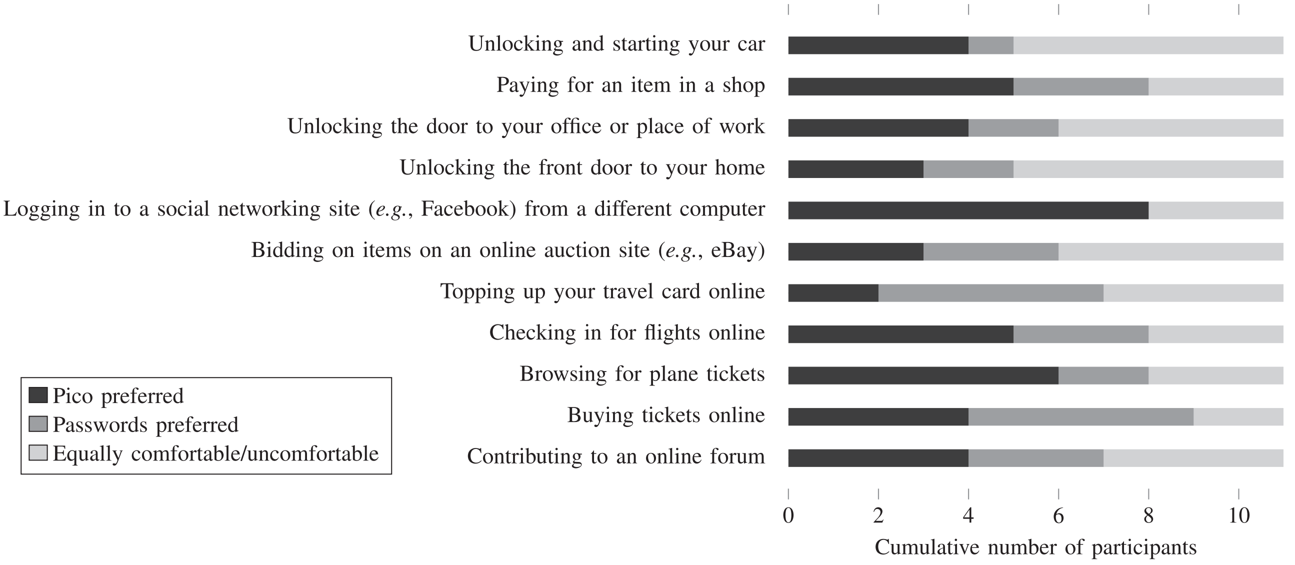


Figure 5: Gyazo study responses showing the numbers of participants who would prefer using Pico, passwords or indifferent for different everyday activities ($N_7 = 11$)

popular were pattern lock and fingerprint scanner, with six mentions each (Table 10).

Password management strategies

When asked about their current password management strategies, three of the $N_9 = 5$ interviewees said they used password managers and three reused the same passwords for multiple systems.

I05 reflected on the fact that some accounts were more valuable than others as they guarded access to other things. They explained: '*for Google especially, it has to be incredibly secure because you're asking that to then be responsible for everything else; it's like putting your stuff in a safety deposit box in a bank where you don't trust the people running the bank*'.

Contexts of use

We showed our respondents a series of eleven activities for which they would normally authenticate, and asked them to choose if they would rather use Pico, passwords or if they were indifferent in these situations (Figure 5). While respondents' preferences tend to be evenly distributed between Pico, passwords and indifferent, no respondent indicated passwords as their preferred choice for logging in to a social networking site (e.g. Facebook) from a different computer.

In the interviews, I03 explained Gyazo was a low-security context: '*the chances of someone stealing your phone and trying to log in [to] your Gyazo is pretty low, and especially because there's nothing on it, you can just cheat the pictures and take pictures on that account*'. Apart from the value of the account, frequency of use was a consideration our interviewees mentioned. While one interviewee felt passwords are more suitable for less frequent activities, two argued passwords would instead be better for more frequent ones, saying: '*it's easy for me to memorise passwords that I frequently have to log in to, because that's just how you memorise, is through use*' (I05). The interviewee further explained where Pico would be a suitable authentication method: '*say it's that you only log in to once a month, maybe it's to pay your car insurance or something or a forum that you don't go on very frequently. I could see [Pico] being really handy because it keeps it secure [. . .] instead of having to make a password for each and every single unique obscure thing that you do, you know that this, that the QR code gives you a level of security*'.

Impact of reverse proxy on Gyazo customers

When developing the reverse proxy (Section 'Website login') we were confident that it would be transparent to the end-user, but it was reassuring that none of the participants claimed to have any difficulty understanding the login procedure using Pico.

Given the more involved protocol, and use of client-side Javascript, we were less certain that the implementation would run sufficiently quickly to satisfy user requirements. Once again we were pleased with the results. The average time taken for our protocol to complete was 3.33 s—a small proportion of the overall time users spent during the authentication process, which averaged to 47.5 s. The opinion of the $N_9 = 5$ participants who were interviewed differed on whether they perceived Pico to be faster (two participants) or slower (two participants) than passwords.

Discussion: the pain of passwords on the web

What worked? What went wrong?

Analysis of the results we collected was at face value encouraging in that, for example, a good proportion of respondents tended to agree with positive statements about Pico (Fig. 3), rated the ability to use Pico on other sites as the best potential improvement (Fig. 4) and preferred passwords to Pico in only 2 of the 11 envisaged scenarios (Fig. 5).

On the whole, however, we cannot hide our disappointment at the fact that, despite having successfully partnered with a website boasting $N_0 > 9$ million monthly active users, we managed to get only $N_6 = 11$ of them to use Pico to log in, which removed any hope of quantitative statistical significance for our results.

This gave us something to reflect on. Even just the drop from $N_0 > 9$ million users of Gyazo to $N_1 = 1,136$ users who had logged in at least once in the previous week was a strong signal that most users of Gyazo didn't actually experience much of a problem with passwords. Gyazo, like most successful websites that want to retain their users, proactively removes any barriers that might prevent users from accessing their accounts, for fear that they might vote with their feet and stop using the site. Florêncio and

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

Herley [12] insightfully remarked back in 2010 that commercial sites that must compete vigorously for user traffic have much more lenient password requirements than government and university sites that enjoy a captive audience. Effectively, Gyazo had already practically solved the password problem for its users in over $(N_0 - N_1)/N_0 = 99.99\%$ of the cases, firstly by not requiring a login for the commonly used snapshot capture functionality but just for the relatively less frequent operation of accessing your in-cloud pictures; and secondly by installing long-lived login cookies that meant you could usually access your Gyazo web account without having to retype your password.

It is significant that the few users we could debrief told us that they would have liked Pico to remember *all their other* web passwords. Their password pain on the web came not from having to type their Gyazo password too often, but from having to keep track of dozens or hundreds of infrequently used accounts. This is a valuable insight: the ‘pain of passwords on the web’ has largely been mitigated by long-lived login cookies and in-browser password managers, and most of the residual pain, on the web, is in accessing infrequently used accounts, perhaps not from your usual computer.

Pivoting to a new scenario

The clear message from this Gyazo study, which in hindsight we could have spotted even at Stage 1, was therefore that we failed because we were trying to offer a remedy to people who were not experiencing enough pain. In order for Pico to be something that people eagerly demanded, rather than just tried out of curiosity, it had to address a situation in which users definitely felt a much greater amount of password-induced pain.

After some brainstorming, we pivoted to the scenario of employees who had to unlock their corporate computer. Because they were employees, rather than customers, they would probably be subject (as indicated by the previously cited Florêncio and Herley [12]) to the more draconian password requirements that system administrators feel entitled to impose when they have a captive audience; and, because they would not yet be logged into their machine at the time of entering their password, they could not be exonerated from typing the password by a long-lived cookie or a password manager. On the contrary, the corporate policy might even mandate automatic locking after half an hour of inactivity, forcing them to retype their password several times a day. This auto-locking policy, common in corporate environments, is particularly annoying to users. We figured that, if Pico relieved users from that burden, it could become very popular. With continuous authentication we would be unlocking the computer simply by walking up to it, not by scanning a QR code; and locking it again automatically by walking away. By offering continuous authentication, Pico would improve on both the usability (no more password retyping) and the security front (the computer locks as soon as you leave your desk, not half an hour later).

Of course, a moment’s thought shows that reality can be much more complicated, for example when there are several computers in the room on which one has an account. But the potential benefits were sufficiently compelling that our team committed to supporting this feature.

The proximity-based laptop-unlocking study with Innovate UK

Background

About Innovate UK

Innovate UK is a hi-tech government agency that supports business-led innovation. After initial engagement with their management and their IT department, they generously agreed to run a live trial of the new version of Pico we had chosen to develop. As an innovation-oriented outfit they understood all too well that alpha and beta versions would have rough edges, but wanted to give us a chance to iron them out by offering feedback as a friendly ‘client zero’. A useful piece of software might one day turn into a successful business and encouraging such endeavours aligned well with their mission.

Table 11: an overview of the number of people during the different stages of the Innovate UK study

Number of people		Description of stage
N_{10}	**1**	**Just the IT manager evaluating Pico** (Stage 1)
N_{11}	11	IT team members suggested by the IT manager
N_{12}	6	Subset willing to take part in trial
N_{13}	**3**	**Actually used Pico and sent us feedback** (Stage 2)
N_{14}	Over 100	Employees in the organization
N_{15}	About 50	Might have wanted to take part in trial (but we never got to Stage 3)

Their employees had access to a variety of computing platforms but we were told that their standard issue laptop was a Lenovo Miix 720 running Windows 10, whereas their most common smartphone was an Apple iPhone 7 running iOS 11. We were asked to support those two platforms at a minimum. This involved (besides procuring instances of those exact devices) a substantial rewrite of our codebase, because our previous internal attempts at OS login with Pico had instead targeted the Ubuntu Linux desktop and Android smartphones.

Our experiment

The IT manager at Innovate UK was extremely supportive of our endeavours but he was also a realistic person. While he was happy to run pre-production software and support our experiments, he did not want to inflict on his users a fragile system that did not meet some minimum stability requirements. We therefore agreed on a staged deployment (Table 11). In Stage 1, we would initially deploy a feature-restricted version of Pico just to him ($N_{10} = 1$), receive his feedback, address any problems he experienced and add any features he deemed necessary. Once he considered the software sufficiently ready, for Stage 2 he would allow it to be deployed, on an opt-in basis, to selected members of his system administration team. He suggested $N_{11} = 11$ people, whom we invited, and $N_{12} = 6$ of them accepted; but only $N_{13} = 3$ of them were using compatible devices and were thus able to engage fully with the trial, providing detailed feedback and logs.

The plan was that, once the Stage 2 participants also gave their OK, in Stage 3 Pico would eventually be made available, on a voluntary basis, to all employees of the organization ($N_{14} = 100+$). The expectation was that about $N_{15} = 50$ of them might want to run the Stage 3 trial, which would last about a month. In terms of data collection we planned to conduct initial interviews, surveys throughout the trial and final debriefing interviews with at least some of the participants. We agreed with the IT manager that the mark of Pico’s success would be if Innovate UK employees spontaneously demanded to continue to use it at the end of that trial month.

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

Unfortunately, we struggled with the reliability of Bluetooth and we never went beyond Stage 2 of our plan. The data that we report come from an initial interview with the IT manager, from emails, from conversations on the phone and from face to face meetings during three site visits. We never sent out any surveys as we felt our software had not yet reached a sufficiently stable state.

As for demographics, we did not get to the stage where we could have selected a balanced mix of participants representative of a broader population. For the record, the $N_{13} = 3$ participants of Stage 2, which included the lone participant of Stage 1, were all male and (as members of the IT team) highly IT literate.

Experimental methodology

Data collection and analysis

Our quantitative data are limited because, given our ideological commitment to privacy, login data were initially stored only on the user's phone. Participants had the option of sending the logs to us through the Pico app, but they usually did so only when they experienced a problem. We received about 30 such log files, totalling about 600,000 log entries. We were also able to capture and look at debug data from the iPhone and Windows software in real time during our site visits to Innovate UK. This allowed us to inspect program execution state as well as the packets transmitted between the devices.

We analysed the data using Braun and Clarke's thematic analysis [10] and focused on pivotal events (e.g. first encounter).

Research ethics

The study never went beyond the scoping stage (Stage 2), so we did not apply for an ethics approval with our university. However, we were cognisant of the potential security implications of working with real users as part of their real work routine. User privacy and security were therefore our highest priority.

The software was developed with security and privacy in mind. We then also contracted an independent penetration testing company to perform a whitebox security audit before deploying Pico to users.

The trial was entirely on an opt-in basis without any pressure or incentive to participate. All of the participants were both IT and security literate and were able to make their own risk assessment. They were in fact assessing a range of different security products at the same time as our trial and this formed part of their work.

No analytics were transferred from the devices, but users did have the option to send us logs, which they were also able to view themselves before sending. We collected some data directly from the device in the presence of the users concerned (a necessary restriction because admin access was needed in some cases, and we neither had access to user passwords nor wanted it).

Although the logs and onsite analysis captured data transmitted between the phone and tablet over the Bluetooth link, only ciphertext was captured, without the keys needed to decrypt it. This allowed us to establish timing and size of packets, but not the content.

Trial diary

The working prototype of OS login using Pico that we had when we originally contacted Innovate UK used an Android phone to log in to the Ubuntu Linux desktop and therefore, before our trial could even be approved by Innovate UK management, we first had to invest considerable development effort in porting the client side to iOS and the server side to Windows. In this section, we chronicle the salient points of our development effort.

The 'continuous authentication' functionality we planned to test was heavily reliant on recent BLE capabilities, specifically a continuous BLE connection being maintained between the user's phone and their laptop. A ping-pong communication was triggered by the phone every two seconds and, if the laptop failed to receive a communication within five seconds, the laptop would lock. The locking functionality therefore relied on reliable and timely communication between the phone and laptop. On deployment, we were aware of some situations where this system would fail and the connection would drop out. For example, the Windows Credential Provider was set to restart every 30 s and, if a connection was made in the small window of time during which it was restarting, the authentication request would be refused.

After initial deployment, it became clear that unintended connection failures like this—but with different root causes—were happening rather more frequently than anticipated. This had the effect of locking the user's computer, often for no apparent reason and while the user was in the middle of doing something. This was unacceptable, so we focused heavily on addressing this issue.

The greatest challenge we encountered was that we were unable to fully replicate all of the failures on our own devices. As an example, some of the messages sent from the iPhone would arrive corrupted. We weren't able to replicate the behaviour, but we could see the effect clearly in log files extracted from a phone on-site. We had initially chosen not to collect the actual data transmitted in the logs, just the timestamps; but no obvious pattern emerged from the timing of the failures. Thus it wasn't until collecting more detailed logs in the presence of the user concerned, and using his devices, that we noticed that the corruption happened when the message size exceeded 254 bytes. But we still didn't know *why* it happened, and it still wasn't happening on our own identical hardware.

As a result of this and other unresolved technical issues, we changed the functionality to require user intervention. If the user moved away from their tablet or laptop, they would be prompted with a notification to lock the device. Clicking the notification on their phone would lock the device remotely, but the fully automatic lock was removed to prevent the machine from locking unexpectedly in mid-use.

We also discovered that user perceptions of security (as opposed to actual security) were far more important than we had anticipated. For example, our software was criticized because it was unsigned, leading to a warning being displayed by Windows on installation.

> '*…when attempting to install it I was getting this system notification* [*Windows unknown publisher warning*]. *It would put other users off. Is there a way this can be circumvented?*' (P01)

To address this, we purchased a code-signing certificate, whereas in a lab trial we might have simply asked the users to ignore the warning, or manually installed the software on the user's behalf.

It wasn't until week 25 that we finally graduated to Stage 2 and offered the app for use by more people within the organization. As with the Gyazo trial, we had to reduce our expectations in terms of numbers. Although the app was offered to $N_{11} = 11$ people from within IT and tech support, five of them chose not to engage. Of the remaining $N_{12} = 6$, it immediately became clear that three would not be able to use the app because they used macOS on their laptops rather than Windows, or Android on their phones rather than iOS. One of the $N_{13} = 3$ remaining users immediately experienced difficulties using Pico with multiple accounts and with third-party app-sandboxing software:

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

> '*When pairing to mobile the pico app is populating my admin credentials to pair – I am trialling* [*3rd party software*] *at the moment and as a result my personal AD* [*Active Directory*] *account does not have administrator access, so I need admin credentials to download/install/run programs.*' (P02)

The final user in the trial experienced a gap between the expectation and reality of the app.

> '*Sorry for sounding so negative about this product, I am sure with some work it could be really good. I was hoping for something that would detect when you are outside of Bluetooth range, then lock the machine when signal got weak. This APP still requires manual intervention, which means users may as well manually lock their laptop.*' (P03)

Indeed, our original vision for how Pico ought to work matched what this user expected; but, as we said, we experienced false positives in the out-of-range detection that caused users to be locked out while they were still sitting at their computer. To avoid that, we had to add an inelegant manual confirmation.[4]

We also found that functionality that we thought would be important for users, and that we spent some time developing, turned out not to be as important as we had anticipated. For example, the app was developed to handle multiple logins to multiple devices with a UI designed to allow selection between them. In practice, users were pairing with only a single device, and found that the extra UI complexity added unnecessary steps to the login process.

Lessons learnt

We found that, for authentication, reliability is key. The main roadblock for us in getting the hoped-for results in a live trial was the difficulty of achieving a level of reliability that would not have been needed for a lab-based or 'simulated' trial. Even if participants in lab-based studies complain about accuracy and reliability of security software (e.g. as shown by Krol *et al.* [13] for facial biometrics), they do not suffer the real-world consequences of not being able to carry on with their primary task.

We were also necessarily very cautious about product security, but also arguably over-protective of user privacy. As a result, we did not collect enough data to be able to reproduce the problems that occurred. This was not a consequence of the low number of users involved, but rather of the approach chosen to collect data. Passively collecting data as we had done in the Gyazo study would have been a far more effective approach, in contrast to having participants actively choose to send us logged data.

A clear piece of feedback received was that the visual appeal of the app was important:

> '*I think some work is needed with the app's GUI. The pairings menu is functional, but not exactly graphically pleasing. I am wondering also whether you should have two app buttons, one for the admin features and the other as a shortcut so that you don't have to click on the app, open the pairings screen and then select the device you'd like to pair. If this could be done by simply clicking on a shortcut on a mobile it would make it more attractive from a user's perspective.*' (P01)

User expectations in this area have become very high, and while in a lab environment users may be persuaded to look past the presentation of an app, this is unlikely to be the case if they're asked to use it in their day-to-day activities. This has particular relevance for security solutions, since the substance (that is, whether or not the solution is secure) is hidden, and it is inevitable that users will therefore focus on other indicators.

The phenomenon of associating visual appeal with functional quality has been demonstrated by usable security research. In their study on phishing, Wu *et al.* [14] showed that participants could be tricked into submitting personal information 34% of the time. Although participants were instructed to pay attention to the toolbars in the browser, they assessed the legitimacy of the websites they visited by how the web pages looked and felt. In a study into two-factor authentication for online banking [15], a participant preferred a one-time password generator that looked more stylish although the underlying functionality and usability was the same.

Limitations of our studies

The numbers of participants in both studies are small. We can extract anecdotes and insights from them but we cannot extrapolate to the general population.

In both studies, we had a low number of observations, meaning login events. Pico did not become part of the users' routines and we could not observe habituation.

The drastic reduction in N_i as we progressed through the stages of the first study might have caused self-selection bias in the participant group.

In the Gyazo study, participants tended to be younger. In the second study, they were IT professionals. Both groups might have higher levels of computer literacy than the general population.

We did not have a control group. However, in the Gyazo study, we secured a baseline against which we could compare Pico by capturing participants' experiences with passwords (*cfr.* the questionnaires and interviews discussed in Section 'Perceptions of Pico').

Related work

The literature identifying systemic problems with passwords as an authentication mechanism is well-known and goes back to the 1970s with Morris and Thompson [16], but the quest to replace passwords with better mechanisms has turned out to be a war of attrition. Over the years, numerous password replacement schemes have been proposed and tested, each with benefits that turned out to be too niche to achieve widespread adoption.

Biddle *et al.* [17] survey research into graphical passwords as a replacement for textual passwords. They conclude that studies lack consistency, often failing to achieve rigorous evaluation of security or usability. The evaluation checklist they provide for addressing these failings identifies user studies and ecological validity as an important factor.

The majority of studies evaluating authentication mechanisms have been laboratory-based trials. The mechanisms studied include graphical passwords, by Biddle *et al.* [17], Passfaces, by Davis *et al.* [18] and grids, by Brostoff *et al.* [19] and Krol *et al.* [20] to name a few. In reality, security-related actions are secondary tasks and a

[4] Around the time of our trial, Apple introduced the ability to unlock a macOS computer without a password when in proximity of the owner's unlocked Apple Watch. That was a commercial authentication product that had clearly undergone several more orders of magnitude of hours of development and testing than ours, and with much better hardware and radio debugging facilities, but even they had chosen not to provide automatic locking when the user went out of range, presumably for similar reasons of not wanting to expose customers to frustrating false positives.

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

study has to mimic this setup. Although Beautement and Sasse [21] have been calling for robust authentication studies for a long time now, many studies such as the one by Bonneau and Stechter [22] still rely on simulated interactions in artificial setups, failing to consider how authentication fits in with users' daily activities. If the interactions and logins are not real, the validity of the studies is limited, argue Krol *et al.* [23].

While Felt *et al.* [24] have conducted in-the-wild studies for security warnings, literature about testing authentication mechanisms in the wild is rare. This is not to say that testing itself is rare: we imagine it must be common in industry, but in contexts where the incentives reward perfecting the product rather than writing about it. A notable exception is the work by Brostoff and Sasse [25] who studied Passfaces in a three-month field trial with 34 students. The students had to use Passfaces and passwords to access their course materials. The authors found that, when using Passfaces, participants logged in with a third of the frequency of logging in with passwords because the login process was more time-consuming. Participants also stayed logged in for longer when using Passfaces.

In recent years, several studies have been conducted to assess the user experience of token-based credentials. Most of them looked at technologies that used a token as part of a two-factor authentication solution. Strouble *et al.* [26] studied the introduction of the Common Access Card (CAC) to the US Department of Defense (DoD). The CAC is a smart card and photo ID, which DoD employees use for both opening doors and logging in to computers. The introduction of the CAC had a significant negative impact on organizational productivity: employees reported that the increased difficulty of accessing their emails led them to log in less often when outside their primary workplace. Also, over two thirds of employees inadvertently left their CAC in the computer. The authors estimated this resulted in a productivity loss of $10.4 million. Similarly, Steves *et al.* [27] studied authentication in a large US governmental organization. They found that employees disliked using RSA's SecurID and the elaborate login procedure discouraged them from logging in remotely. Krol *et al.* [15] studied the user experience of authentication tokens for UK online banking. They found that the need to have a hardware token was a source of inconvenience and it changed the way participants went about doing banking, decreasing the frequency of login. Participants reported being less satisfied with online banking when more steps were required for the login process and if they had to use a hardware token. UK banks have since been shifting from physical to software tokens to relieve their customers of the burden of carrying an additional device for generating a one-time password. A preference for phone-based authentication has been documented in real-life deployments. Colnago *et al.* [28] followed the introduction of 2-factor authentication at Carnegie Mellon University where staff and students had the choice between different methods for generating passwords. The results show that 98% chose to use their smartphone for Duo authentication. The three most frequent setups users had were: push notification (91%), app-generated OTP (21%) and hard token (4%).

There has also been a body of work focusing on making authentication continuous. In 1992, Want and Hopper [29] proposed the *Active Badge* that would log the user in to their workstation if they were in proximity and check for their presence at several points during a session. In 1997, Landwehr [30] proposed, patented (with Latham [31]) and implemented a more robust approach—a system that would continuously monitor the presence of an RFID token worn by the user and, in its absence, disconnect the keyboard and monitor from the computer. To address the vulnerability that an attacker could still access the disk, in 2002 Corner and Noble [32] presented (and later refined with Nicholson [33]) 'Zero-Interaction Authentication'. They implemented a system in which proximity of an authentication token would unlock the keys of the laptop's cryptographic file system. The absence of the token would flush the decrypted files from the cache and erase their decryption keys.

Another way to alleviate the password burden has been single sign-on (SSO), in which a single master account (and therefore only one password) provides access to multiple other services. The work by Pashalidis and Mitchell [34] provides a useful taxonomy of the various SSO systems based on two dimensions: local *versus* proxy (does the entity the user authenticates to reside on the local computer or in the network?) and true *versus* pseudo (is SSO supported by design, or do several systems accept the same password?). Research into usable security has highlighted several shortcoming of real-world implementations of SSO. Linden and Vilpola [35] showed that, when using SSO, users were unaware if they are logged in or not. For example, some of their participants thought they logged out of all systems, while in reality they logged out of only one. This raises security and privacy concerns. Inglesant and Sasse [36] showed that in an organization that used a pseudo-SSO implementation, employees were frustrated by the need to enter the same password separately for every system and by the frequency of lockouts. Ruoti *et al.* [37] found that users have concerns about using one password for many systems, and in the case of federated identity the trust towards the identity provider plays a role.

Bonneau *et al.* [38] identified three overarching classes of benefits that an authentication mechanism should provide: usability, deployability and security. In this study, we assess Pico's usability, deployability and perceived security. Analysis of the empirical usability of a single-factor token-based password replacement that is *Memorywise-Effortless* and *Physically-Effortless* (in the jargon of Bonneau *et al.* [38]) is an under-explored area that would still benefit from additional contributions.

Conclusions

We conducted the first two studies of a functional implementation of Pico: one with the users of a website, Gyazo, engaging in remote authentication across the Internet; and another with the employees of an organization, Innovate UK, engaging in local login to their laptops. Both studies took place in real environments and stand out for their emphasis on high ecological validity.

In the Gyazo study, we sought to understand how useful Pico might be as a replacement for passwords on a major website. The insights we gleaned by debriefing our test subjects are valuable, but hard to generalize given the small sample size.

Arguably, the most valuable lesson from this study is at the meta-level: why was it that, on a site with over 9 million active users, we managed to get only 11 people to try Pico? We deduced that website users do not feel the pain of passwords as strongly as some security usability researchers think they do.

As we said in the original workshop write-up of the Gyazo study [9], we intended to use our findings to shape the future development of Pico. In this article, we show that we did. We pivoted to a different scenario in which the pain of passwords would be felt much more strongly, where people would have to type a complex password several times a day and could not be excused from typing it by a password manager or a long-lived cookie. We didn't care so much about publishing another paper, but we really cared about applying Pico to a setting in which it could solve a concrete and painful password problem that people were experiencing.

In the Innovate UK study, however, despite our best efforts, we could not make our implementation sufficiently robust in the face of

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

unexplained and hard-to-reproduce Bluetooth dropouts. This regrettably prevented us from deploying Pico at scale to the corporate audience we had originally envisaged.

Many usability studies in the literature are based on online surveys or laboratory experiments where prototypes are used under controlled conditions. If we had stuck to that paradigm, it would have been much easier to collect hundreds of responses and derive statistically significant results that would have looked good in scientific publications, whereas by insisting on piloting our system with actual users who go about their regular workday we ended up being able to engage with only $N_7 = 11$ of them in the Gyazo study and $N_{13} = 3$ in the Innovate UK study. Yet we defend and advocate this more challenging, more expensive and more frustrating approach as the one that leads to results of much greater ecological validity. While it is appropriate to test early ideas inexpensively in constrained, simulated environments such as the web survey or the lab experiment with one's own undergraduates, there remain major qualitative differences between those settings and the real-world ones in which the system would eventually have to operate. Looking for answers by conducting a study on Amazon Mechanical Turk[5] rather than through deploying the technology to actual users is cheaper, simpler and more scalable, but may be akin to looking for lost keys under the lamp post, because that is where it is easier to see, rather than in the long, dark and smelly alleyway where they may have actually been dropped.

We invested a considerable number of person-months of development and debugging effort into the Innovate UK trial but, in the end, we couldn't get our system to work for our 'client zero' and we don't have any pretty graphs to show for it. Yet, we argue that reporting this negative result is a more scientifically honest and ultimately more valuable assessment of our system than a hypothetical lab study smugly reporting that $x\%$ of users had found Pico better than passwords. We believe the academic community puts undue pressure on reporting positive outcomes and this creates perverse incentives. If this is the playing field, it should come as no surprise that a majority of the 35 authentication schemes from the literature examined in the full version of Bonneau *et al.* [40] have never seen any practical use. If only positive outcomes are publishable, and if publications are required to advance a researcher's career, then academics will be steered towards building only as much of a prototype as needed to be able to run a few lab studies, as opposed to first building and debugging a solid working system, and then witnessing how real users would react to it. Clearly, there is a time and place for prototyping ideas cheaply before investing engineering effort into them; but we argue that, once the novel ideas have been explored conceptually, the true answers about their worth can only be obtained by building a working system and letting people use it in their daily life. Until we do this, our lab studies are not validated by reality. Talented young researchers, especially those without a tenured position, should not be deterred from engaging in solid system work as a foundation to their usability studies by the fact that in the same time they could produce more papers if they only built and tested the Hollywood façade instead.

While the methodology of design / build / deploy / test / refine is not novel in industry, in academic research it seems to have been adopted primarily in hard-core technical areas, such as building compilers and operating systems. In the emerging domain of studies on the usability of security, instead, it is unfortunately still rare to see the above cycle extend all the way to an end-to-end testing of a working prototype used by non-technical people in their ordinary workday. We do not claim that this end-to-end approach is a new discovery, nor that nobody adopted it before us in this context; but we argue that the field of security usability is now mature enough that its practitioners would generally benefit from stepping up to this higher standard.

Supplementary data

Supplementary data is available at *Journal of Cybersecurity* online.

Acknowledgements

We are grateful to our colleagues David Harrison, Robert Irvine, Jeunese Payne, Max Spencer, Quentin Stafford-Fraser, Matthew Wahab and Chris Warrington for their contributions to the Pico project as well as to our external collaborator Jiny Bradshaw whose Bluetooth know-how was invaluable. We are particularly grateful to Nota Inc., the parent company of Gyazo, for allowing us to integrate Pico into their site and recruit their users into our study, and to Innovate UK for patiently trialling many versions of Pico over several months in the shared hope that we would at some point nail down those reliability issues. The Gyazo pilot study was originally written up as a workshop paper by the same authors [9] and praised by the referees for its emphasis on ecological validity. We were subsequently invited by the guest editors to extend that work and resubmit it to this journal. The Innovate UK pilot study was not previously written up anywhere else. Thanks to Paul van Oorschot for insightful comments on a draft of this article.

Conflict of interest statement. None declared.

Funding

This work was supported by the European Research Council (ERC) [StG 307224, Pico, to F.S.] and the extended visits of Gyazo inventor and CTO T.M. to Cambridge were supported by the Engineering and Physical Sciences Research Council (EPSRC) [EP/M019055/1, Future authentication systems, to F.S.].

References

1. Stajano F. Pico: No more passwords! In *Proceedings Security Protocols Workshop 2011*, ser. LNCS, B. Christianson *et al.* (eds), Vol. 7114. Springer, March 2011, pp. 49–81. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2011-Stajano-pico.pdf (22 June 2020, last accessed).
2. Jenkinson G, Spencer M, Warrington C *et al.* I bought a new security token and all I got was this lousy phish—Relay attacks on visual code authentication schemes. In *Proceedings of Security Protocols Workshop 2014*, ser. LNCS, B. Christianson *et al.* (eds), Vol. 8809. Springer, April 2014, pp. 197–215. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2014-JenkinsonSpeWarETAL-phish.pdf (22 June 2020, last accessed).
3. Krawczyk H. Sigma: The 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE-protocols. In *Advances in Cryptology – CRYPTO 2003, International Cryptology Conference, Santa Barbara, CA, August 17-21, 2003, Proceedings*, ser. LNCS, Vol. 2729. Springer, 2003, pp. 400–25, invited paper. [Online]; https://iacr.org/archive/crypto2003/27290399/27290399.pdf (22 June 2020, last accessed).
4. Stajano F, Jenkinson G, Payne J *et al.* Bootstrapping adoption of the Pico password replacement system. In *Proceedings of Security Protocols Workshop 2014*, ser. LNCS, B. Christianson *et al.* (eds), Vol. 8809. Springer, April 2014, pp. 172–86. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2014-StajanoJenPayETAL-bootstrapping.pdf (22 June 2020, last accessed).

[5] The well-known crowdsourcing website as described, for example, by Crowston [39].

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020

5. Stajano F, Spencer M, Jenkinson G *et al.* Password-manager friendly (PMF): Semantic annotations to improve the effectiveness of password managers. In *Proceedings of Passwords 2014*, ser. LNCS, Vol. 9393. Springer, 2014, pp. 61–73. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2015-StajanoSpeJenSta-pmf.pdf (22 June 2020, last accessed).
6. Stajano F, Christianson B, Lomas M *et al.* Pico without public keys. In *Proceedings of Security Protocols Workshop 2015*, ser. LNCS, Vol. 9379. Springer, 2015, pp. 212–23. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2015-StajanoChrLomETAL-public.pdf (22 June 2020, last accessed).
7. Stafford-Fraser Q, Stajano F, Warrington C *et al.* To have and have not: Variations on secret sharing to model user presence. In *Proceedings of UPSIDE workshop of UBICOMP*, Seattle, WA, 2014, pp. 1313–1320. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2014-StaffordfraserStaWarETAL-presence.pdf (22 June 2020, last accessed).
8. Payne J, Jenkinson G, Stajano F *et al.* Responsibility and tangible security: Towards a theory of user acceptance of security tokens. In *Proceedings USEC 2016*, February 2016, p. (10 pages). [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2016-PayneJenStaSasSpe-tokens.pdf (22 June 2020, last accessed).
9. Aebischer S, Dettoni C, Jenkinson G *et al.* Pico in the wild: Replacing passwords, one site at a time. In *Proceedings European Workshop on Usable Security (EuroUSEC 2017)*, 2017. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2017-AebischerDetJenETAL-site.pdf (22 June 2020, last accessed).
10. Braun V, Clarke V. Using thematic analysis in psychology. *Qualitative Research in Psychology* 2006;**3**:77–101. https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa.
11. Fleiss JL, Levin B, Paik MC. *Statistical Methods for Rates and Proportions*. Hoboken, NJ: John Wiley & Sons, 2013.
12. Florêncio D, Herley C. Where do security policies come from? In *Proceedings of SOUPS 2010*. ACM, 2010, pp. **10**:1–10:14. [Online]; Available: https://research.microsoft.com/pubs/132623/WhereDoSecurityPoliciesComeFrom.pdf (22 June 2020, last accessed).
13. Krol K, Parkin S, Sasse MA. Better the devil you know: A user study of two CAPTCHAs and a possible replacement technology. In *NDSS Workshop on Usable Security (USEC 2016)*, 2016. [Online]; https://discovery.ucl.ac.uk/id/eprint/1535331/ (22 June 2020, last accessed).
14. Wu M, Miller RC, Garfinkel SL. Do security toolbars actually prevent phishing attacks? In *Conference on Human Factors in Computing Systems (CHI)*, Montréal, Québec, 2006, pp. 601–10.
15. Krol K, Philippou E, De Cristofaro E. *et al.* 'They brought in the horrible key ring thing!' Analysing the usability of two-factor authentication in UK online banking. In *NDSS Workshop on Usable Security (USEC)*, 2015. [Online]; https://discovery.ucl.ac.uk/id/eprint/1461425/ (22 June 2020, last accessed).
16. Morris R, Thompson K. Password security: A case history. *CACM* 1979; **22**:594–97.
17. Biddle R, Chiasson S, van Oorschot PC. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys (CSUR)* 2012;**44**:1–44.
18. Davis D, Monrose F, Reiter MK. On user choice in graphical password schemes. *USENIX Security* 2004;**13**:11.
19. Brostoff S, Inglesant P, Sasse MA. Evaluating the usability and security of a graphical one-time PIN system. In *BCS Interaction Specialist Group Conference*, Dundee, Scotland, 2010, pp. 88–97. British Computer Society.
20. Krol K, Papanicolaou C, Vernitski A *et al.* 'Too taxing on the mind!' Authentication grids are not for everyone. In *Human Aspects of Information Security, Privacy, and Trust (HAS), HCI International 2015*, Vol. LNCS 9190, 2015, pp. 71–82. [Online]; https://discovery.ucl.ac.uk/id/eprint/1470881/ (22 June 2020, last accessed).
21. Beautement A, Sasse MA. Gathering realistic authentication performance data through field trials. In *Usable Security Experiment Reports (USER) Workshop at the Symposium on Usable Privacy and Security (SOUPS)*, 2010.
22. Bonneau J, Schechter SE. Towards reliable storage of 56-bit secrets in human memory. In *USENIX Security*, San Diego, CA, 2014, pp. 607–23.
23. Krol K, Spring JM, Parkin S *et al.* Towards robust experimental design for user studies in security and privacy. In *The LASER Workshop: Learning from Authoritative Security Experiment Results (LASER)*. IEEE, 2016. [Online]; https://discovery.ucl.ac.uk/id/eprint/1503240/ (22 June 2020, last accessed).
24. Felt AP, Reeder RW, Almuhimedi H *et al.* Experimenting at scale with Google Chrome's SSL warning. In *Conference on Human Factors in Computing Systems (CHI)*. ACM, 2014, pp. 2667–70. [Online]; https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41927.pdf (22 June 2020, last accessed).
25. Brostoff S, Sasse MA. Are Passfaces more usable than passwords? A field trial investigation. *People and Computers XIV – Usability or Else!*, pp. 405–424, 2000. [Online]; https://discovery.ucl.ac.uk/id/eprint/19830/ (22 June 2020, last accessed).
26. Strouble DD, Schechtman G, Alsop AS. Productivity and usability effects of using a two-factor security system. In *Annual Conference of the Southern Association for Information Systems*, 2009.
27. Steves M, Chisnell D, Sasse MA *et al.* Report: Authentication Diary Study. National Institute of Standards and Technology (NISTIR) 7983, 2014. [Online]; https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7983.pdf (22 June 2020, last accessed).
28. Colnago J, Devlin S, Oates M *et al.* 'It's not actually that horrible': Exploring adoption of two-factor authentication at a university. In *Conference on Human Factors in Computing Systems (CHI)*. ACM, 2018, p. 456. [Online]; http://www.contrib.andrew.cmu.edu/~nicolasc/publications/Colnago-CHI18.pdf (22 June 2020, last accessed).
29. Want R, Hopper A. Active badges and personal interactive computing objects. *IEEE Transactions on Consumer Electronics* 1992;**38**:10–20. [Online]; https://dl.acm.org/doi/10.1109/30.125076 (22 June 2020, last accessed).
30. Landwehr CE. Protecting unattended computers without software. In *Annual Computer Security Applications Conference*. Washington, DC, USA: IEEE Computer Society, December 1997, pp. 274–83. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a465472.pdf (22 June 2020, last accessed).
31. Landwehr CE, Latham DL. Secure identification system. US Patent 5,892,901, filed 1997-06-10, granted 1999-04-06, 1999.
32. Corner MD, Noble BD. Zero-interaction authentication. In *Procedings ACM MobiCom 2002*, 2002, pp. 1–11. [Online]; https://www.sigmobile.org/mobicom/2002/papers/p002-corner.pdf (22 June 2020, last accessed).
33. Nicholson A, Corner MD, Noble BD. Mobile device security using transient authentication. *IEEE Transactions on Mobile Computing* 2006;**5**: 1489–502. [Online]; https://people.cs.umass.edu/~mcorner/papers/tmc_2005.pdf (22 June 2020, last accessed).
34. Pashalidis A, Mitchell CJ. A taxonomy of single sign-on systems. In Information Security and Privacy, ACISP 2003, ser LNCS, Safavi-Naini R, Seberry J (eds), Vol 2727. Springer, 2003, pp. 249–64.
35. Linden M, Vilpola I. An empirical study on the usability of logout in a single sign-on system. In Information Security Practice and Experience. ISPEC 2005, ser. LNCS, Deng RH *et al.* (eds), Vol 3439. Springer, 2005, pp. 243–54.
36. Inglesant PG, Sasse MA. The true cost of unusable password policies: Password use in the wild. In *Conference on Human Factors in Computing Systems (CHI)*. ACM, 2010, pp. 383–92. [Online]; https://discovery.ucl.ac.uk/id/eprint/102754/ (22 June 2020, last accessed).
37. Ruoti S, Roberts B, Seamons K. Authentication melee: A usability analysis of seven web authentication systems. In *International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Florence, Italy, 2015, pp. 916–926.
38. Bonneau J, Herley C, van Oorschot PC *et al.* The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proceedings IEEE Symposium on Security and Privacy*, 2012, pp. 553–67. [Online]; https://www.cl.cam.ac.uk/~fms27/papers/2012-BonneauHerOorSta-password.pdf (22 June 2020, last accessed).
39. Crowston K. Amazon Mechanical Turk: A research tool for organizations and information systems scholars. In Bhattacherjee A and Fitzgerald B (eds), *Shaping the Future of ICT Research. Methods and Approaches*, ser. IFIP Advances in Information and Communication Technology, Berlin, Heidelberg: Springer, 2012, Vol. **389**.
40. Bonneau J, Herley C, van Oorschot PC *et al.* The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. University of Cambridge Computer Laboratory, Tech Report 817, 2012. [Online]; https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-817.pdf

Downloaded from https://academic.oup.com/cybersecurity/article/6/1/tyaa010/5989371 by guest on 19 November 2020