Passwords and the Evolution of Imperfect Authentication

Joseph Bonneau
Stanford University & EFF
jbonneau@cs.stanford.edu

Cormac Herley
Microsoft Research
cormac@microsoft.com

Paul C. van Oorschot
Carleton University
paulv@scs.carleton.edu

Frank Stajano
University of Cambridge
frank.stajano@cl.cam.ac.uk

Abstract

Theory on passwords has lagged behind practice, where large providers use back-end smarts to survive with imperfect technology. Simplistic models of user and attacker behaviors have led the research community to emphasize the wrong threats. Authentication is a classification problem amenable to machine learning, with many signals in addition to the password available to large Web services. Passwords will continue as a useful signal for the foreseeable future, where the goal is not impregnable security but reducing harm at acceptable cost.

A version of this work appeared in *Communications of the ACM* vol. 58 no. 7, July 2015 pp. 78–87.
This is the author's own version.

1 Introduction

Passwords have dominated human-computer authentication for 50 years despite consensus among researchers that we need something more secure and deserve something more user friendly. Much published research has focused on specific aspects of the problem that can be easily formalized but do not actually have a major influence on real-world design goals, which are never authentication *per se*, but rather protection of user accounts and sensitive data. As an example of this disconnect, academic research often recommends strict password-composition policies (such as length requirements and mandating digits and non-alphabetic characters) despite the lack of evidence that they actually reduce harm.

We argue that critically revisiting authentication as a whole and passwords' role therein is required to understand today's situation and provide a meaningful look ahead. Passwords were originally deployed in the 1960s for access to time-shared mainframe computers, an environment unrecognizable by today's Web users. Many practices have survived with few changes even if no longer appropriate [9, 19]. While partly attributable to inertia, this also represents a failure of the academic literature to provide approaches that are convincingly better than current practices.

We identify as outdated two models that still underlie much of the current password literature. First is the model of a *random user* who draws passwords uniformly and independently from some set of possible passwords. It has resulted in overestimates of security against guessing and encouraged ineffectual policies aimed at strengthening users' password choices. The second is that of an *offline attack* against the password file. This model has inflated the importance of unthrottled password guessing relative to other threats (such as client malware, phishing, channel eavesdropping and plaintext leaks from the back-end) that are more difficult to analyze but significantly more important in practice. Together, these models have inspired an awkward jumble of contradictory advice that is impossible for humans to follow [1, 19, 30, 36].

The focus of published research on clean, well-defined problems has caused the neglect of the messy complications of real-world Web authentication. This misplaced focus continues to hinder the applicability of password research to practice. Failure to recognize the broad range of usability, deployability, and security challenges in Web authentication has produced both a long list of mutually incompatible password requirements for users and countless attempts by researchers to find a magic-bullet solution despite drastically different requirements in different applications. No single technology is likely to "solve" authentication perfectly for all cases; a synergistic combination is required.

Industry has already moved in this direction. Many leading providers have bolstered, not replaced, passwords with multiple parallel complementary authentication mechanisms. These are combined, often using machine-learning, so as to minimize cost and annoyance while providing enough security for e-commerce and online social interaction to prosper. We expect authentication to gradually become more secure and less obtrusive, even if perhaps technically inelegant under the hood.

This trend is not without downsides. It strongly favors the largest providers with extensive knowledge of their users' habits. It makes authentication more privacy invasive and increasingly difficult to comprehend for users and researchers alike. We encourage researchers to acknowledge this trend and focus on adressing related security, privacy and usability challenges.

2 Lessons from the past

2.1 The changing role of passwords

From the beginning, passwords have been a security band-aid. During development of the first time-sharing operating systems in the 1960s, passwords were added to protect against practical jokes and researchers using more resources than authorized. The 1961 Compatible Time-Sharing System at MIT was likely the first to deploy passwords in this manner. Security issues arose immediately. Multiple cases were reported of users guessing one another's passwords and also at least one leak of the master password file that was stored in unencrypted form. Yet these issues were easily adressed administratively

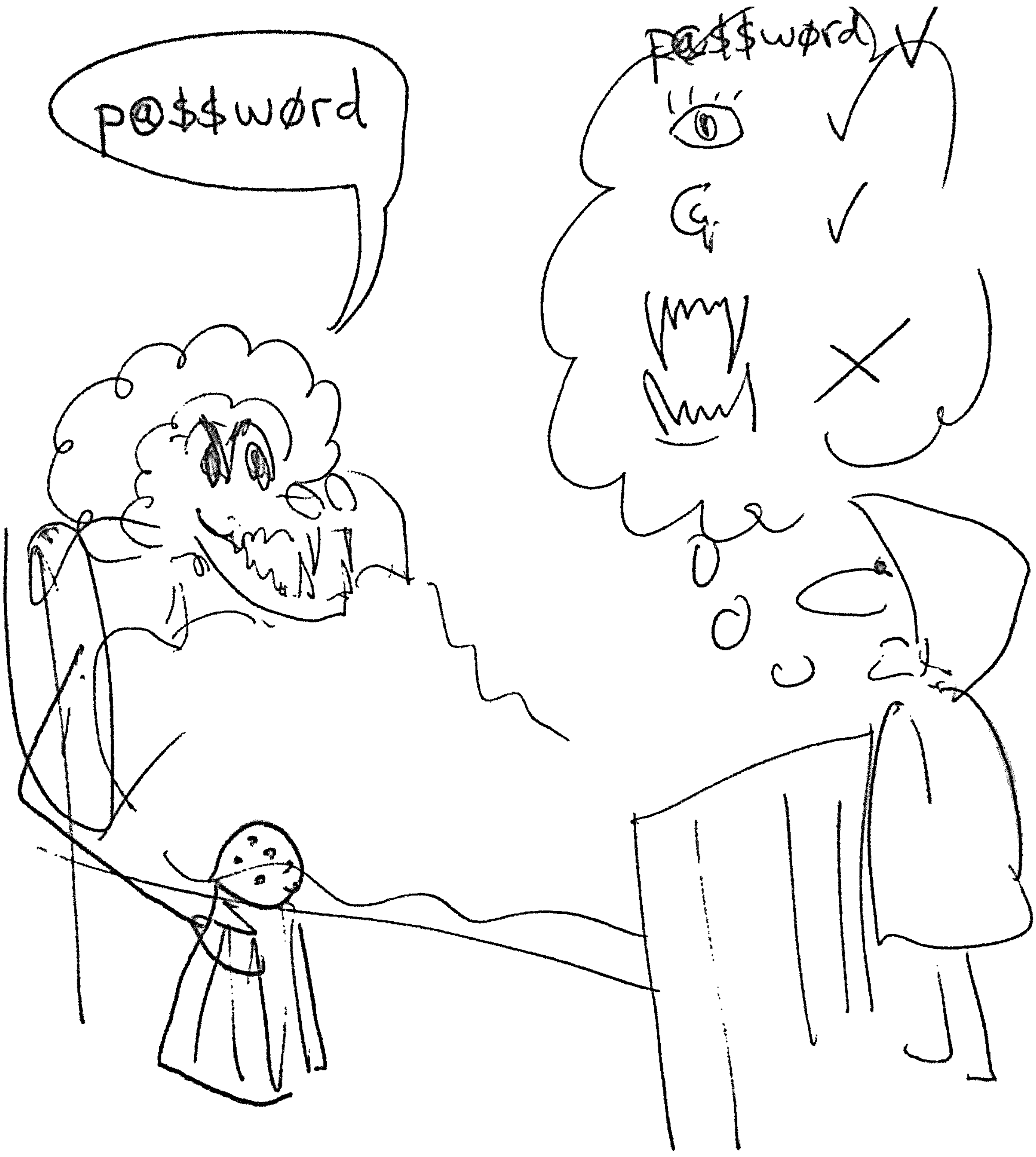

Figure 1: Little Red Riding Hood using multiple signals to authenticate her grandmother. Original sketch by Frank Stajano at IEEE Security and Privacy, May 19 2015.

as all users were part of the same academic organization.

With development of access control in MULTICS and Unix in the 1970s, passwords were adapted to protect sensitive data as well as computational resources. MULTICS protected passwords by storing them in hashed form, a practice invented by Roger Needham and Mike Guy at The Univesity of Cambridge in the 1960s. Robert Morris' and Ken Thompson's seminal 1979 treatment of password security [26] described the evolution toward dedicated password hashing and salting via the `crypt()` function, along with the first analysis of dictionary attacks and brute-force guessing.

A decade later, the 1988 Morris Internet worm demonstrated the vulnerability of many systems to password guessing. Administrators adapted by storing password hashes in more heavily protected *shadow* password files [24] and, sometimes, proactively checking user passwords for guessability [35].

With the mid-1990s advent of the World Wide Web and e-commerce, early attempts were made to replace passwords with public-key cryptography via secure sockets layer (SSL) client certificates or the competing secure electronic transactions (SET) protocol. Ultimately, managing certificates and private keys client-side proved too burdensome and a market never developed. Instead, secure connections on the Web almost universally rely on one-way authenticated SSL. Servers are authenticated by a certificate at the SSL layer, while users are left to prove their identity later with no explicit protocol support. Text-based passwords entered in HTML forms in exchange for HTTP cookies have become the dominant, albeit never formally specified, protocol for user authentication.

As Web-based services proliferated, usability problems arose that had not existed for system passwords. Resetting forgotten passwords, previously a manual task for IT support staff, was automated through email, creating a common central-point-of-failure for most users. The increased number of accounts held by individual users scuttled the assumption of dedicated passwords per account and password re-use became commonplace. Phishing grew into a major concern, but anti-phishing proposals requiring protocol or user-interface changes failed to gain adoption. Instead, the primary countermeasure involves blacklists of known phishing sites and machine learning classifiers to recognize new phishing sites as they arise [16].

Attempts to make a dedicated business out of authentication in the consumer space have failed consistently. While there has long been interest in deploying hardware tokens as a second factor, standalone tokens (such as RSA SecurID) have seen limited deployment outside of enterprise environments, likely due to the cost of tokens relative to the value of free online accounts. Microsoft Passport and OpenID, among many attempts to offer single sign-on for the Web, have failed to gain mass adoption.

The widespread availability of smartphones may be changing the equation, as in the early 2010s a number of online services, including Facebook, Google, and Twitter, deployed free smartphone applications to act as a second factor based on the emerging time-based-one-time-pad (TOTP) standard [27]. Other services send codes via short message service (SMS) as a backup authentication mechanism. A few services have offered dedicated tokens as a second factor, typically in environments at greater risk of fraud (such as eBay or World of Warcraft).

2.2 Random models for user behavior

In addition to being regarded as the weak link in password systems, users are also typically the most difficult component to model. Ideally, they would choose passwords composed of random characters. But even researchers who acknowledge this is an idealized model have usually underestimated the gap between modeled behavior and reality. Security policies and research models have been slow to adjust to the magnitude of the inaccuracies revealed by new data sources (such as leaked password datasets and large-scale measurement studies[5]).

One of the best-known early sources on password policies is the U.S. Defense Department's circa 1985 Green Book [38] which specified detailed policies for mitigating the risk of password guessing, including rate-limiting, hashing passwords at rest, and limiting the lifetime of passwords. The Green Book avoided the complexity of user behavior altogether, putting forth as one of three main principles that: "Since many user-created passwords are particularly easy to guess, all passwords should be machine-generated."

The same year, NIST published its *Password Usage* guidelines in the Federal Information Processing Standards series [39] that were heavily derived from the Green Book. In addition to the recommended machine-chosen passwords, the FIPS guidelines allowed user-chosen passwords with the caveat that users "shall be instructed to use a password selected from all acceptable passwords at random, if possible, or to select one that is not related to their personal identity, history or environment." Today, nearly all non-military applications allow user-chosen passwords due to the perceived difficulty of remembering machine-chosen passwords. Yet the FIPS guidelines retained most other recommendations from the Green Book unchanged, including calculations of password security based on allowed characters and length requirements, limits on password lifetime and forced updates. This encouraged the unrealistically optimistic assumption that users choose passwords similarly to random password generators that has persisted to this day.

2.2.1 Estimating password strength via "entropy"

The guessing resistance of user-chosen passwords is often estimated by modeling passwords as random choices from a uniform distribution. This enables straightforward calculations of expected guessing times in the tradition of the 1985 Green Book. To attempt to capture the fact many users choose from a relatively small number of common passwords (despite the very large theoretical space from which to choose text passwords) researchers often choose a relatively small uniform distribution. The logarithm of the size of this uniform distribution in this model is often called *entropy*, in reference to Claude Shannon's famous measure H_1 of the minimum average number of bits needed to encode symbols drawn from a distribution. Unfortunately, Shannon entropy models the number of guesses needed by an attacker who can check in constant time if an unknown password is a member of an arbitrarily sized set of possibilities. This does not correspond to any real guessing attack.

A more direct metric is *guesswork* (G) [25], the expected number of queries by an adversary guessing individual passwords sequentially until all are found. It can be shown that H_1 provides a lower-bound on G [25]. However, G is also problematic, as it can be highly skewed by rare but difficult-to-guess passwords [5]. To address this bias, *partial* (or *marginal*) guessing metrics have been developed [32, 5]. One formulation is partial guesswork (G_α) which models an attacker making only enough guesses to have a probability α of succeeding [5]. This encapsulates the traditional G when $\alpha = 1$, with lower values of α modeling more realistic attackers. Such metrics have been proven not to be lower-bounded in general by Shannon entropy [32, 5]. While partial-guessing metrics provide an appropriate mathematical model of password-guessing difficulty, they require a very large sample to be estimated accurately, typically millions of passwords [5], and have thus found limited practical use.

Heuristic measures of password strength are often needed for smaller datasets. NIST's Electronic Authentication Guidelines [8] (in many ways an update to the Green Book) acknowledged the mathematical unsoundness of Shannon entropy for guessing but still introduced a heuristic method for estimating "entropy" of password distributions under various composition policies (sometimes called NIST entropy). This model has since been used in many academic studies, although it has been found to produce relatively inaccurate estimates in practice [40, 23]. The preferred empirical approach, albeit dependent on the configuration of a particular tool, is to simply run a popular open-source password cracking library against a set of passwords and evaluate the average number of guesses needed to find a given proportion of them [42, 23]. This approach can be applied to even a single password to evaluate its relative strength, though this clearly overestimates security relative to any real adversaries who use a more favorable cracking library.

2.2.2 Improving password strength

Simple measures like Shannon and NIST entropy make increases in password strength seem tantalizingly close. Composition policies that increase the minimum length or expand the classes of characters a password must contain seem to cause reliable increases in these measures if passwords are random;

for example, the NIST guidelines suggest requiring at least one uppercase and non-alphabetic character. While acknowledging users may insert them in predictable places, they still estimate an increase in guessing difficulty of a password by six bits (or a factor of 64) compared to a policy of allowing any password. However, experiments have shown that this is likely an overestimate by an order of magnitude [40].

Such password policies persist despite imposing a high usability cost (as we will discuss in Section 3.2), though tellingly, their use is far less common at sites facing greater competition (such as webmail providers) than at sites with little competition (such as universities or government services) [14].

Instead, research suggests that the most effective policy is simply using a very large blacklist [34] to limit the frequency of the most common passwords, bounding online guessing attacks to a predictable level, conceding that many users will choose passwords vulnerable to offline guessing.

A related goal has been to nudge users towards better passwords through feedback (such as graphical meters that indicate an estimated strength of their password, as they choose it). In an experimental setting, very aggressive strength meters can make guessing user-chosen passwords dramatically more difficult [37]. However, in studies using meters typical of those found in practice, with users who were not prompted to consider their password, the impact of meters was negligible; many users failed to notice them at all [12]. An empirical data point comes from Yahoo!, where adding a password strength meter did improve password security, but only marginally [5].

2.2.3 Independence when choosing multiple passwords

A random user model often further assumes every password will be independently chosen. In practice, this is rarely true on the Web as users cope with the large number of accounts by password re-use, sometimes with slight modification. For example, a 2007 telemetry study estimated the median user has 25 password-protected accounts but only 6 unique passwords [13]. This has direct security implications as leaks at one website can compromise security at another. Even if a user has not exactly re-used the password, an attacker can guess small variations, which may at least double their chances of success in an online guessing scenario [10]. Related password choices similarly undermine the security goals of forced password updates, as an attacker with knowledge of a user's previous sequence of passwords can easily guess the next password [43].

2.3 Offline vs. online threats

The security literature distinguishes between *online* attackers who must interact with a legitimate party to authenticate and *offline* attackers who are limited only in terms of their computational resources.

Superficially, offline attackers are far more powerful, as they typically can make an unbounded number of guesses and compare them against a known hash of the password. Yet many additional avenues of attack are available to the online attacker: stealing the password using client-side malware, phishing the password using a spoofed site, eavesdropping the password as it is transmitted, stealing the password from the authentication server, stealing the password from a second authentication server where the user has reused it, and subverting the automated password reset process.

A critical observation is that strong passwords do not help against any of these other attacks. Even the strongest passwords are still static secrets that can be replayed and are equally vulnerable to phishing, theft, and eavesdropping. Mandating stronger passwords does nothing to increase security against such attacks.

2.3.1 Offline guessing (cracking)

Much attention has been devoted to devising strategies for picking passwords complex enough to resist offline cracking. Yet this countermeasure may stop real-world damage in at most a narrow set of circumstances. For an attacker without access to the password file, any guessing must be done online, which can be rate-limited. If passwords in a leaked file are unhashed, they are exposed in plaintext regardless of complexity; if hashed but unsalted, then large rainbow tables [31] allow brute-force look-up up to some length.[1] Only if the attacker has obtained a password file that had been properly hashed and salted do password-cracking efficiency and password strength make a real difference.

And yet, while hashing and salting have long been considered best practice by security professionals, they are far from universal. Empirical estimates suggest that over 40% of sites store passwords unhashed [7]; recent large-scale password file leaks revealed many were plaintext (such as Rockyou and Tianya), hashed but unsalted (such as LinkedIn), improperly hashed (such as Gawker), or reversibly encrypted (such as Adobe).

Finally, offline attackers may be interrupted if their breach is detected and administrators can force affected users to reset their passwords. Password resets often are no t instituted at breached websites due to the fear of losing users; they are even less commonly mandated for users whose password may have been leaked from a compromised third-party website where it may have been reused.

2.3.2 Online guessing

Online attackers can verify whether any given password guess is correct only by submitting it to the authentication server. The number of guesses that can be sent is limited. A crude “three strikes” model is an obvious way of throttling attacks but relatively few sites seem to implement such a deterministic policy [7], probably to avoid denial of service.

Nonetheless, online guessing attacks are much more costly to mount than offline ones on a per-guess basis. Whereas, offline, an attacker might check a billion guesses on a single host, online an attacker might need thousands of hosts. If we assume IP addresses that send millions of failed attempts will be blocked, the load must be distributed. Second, the online guess requires an HTTP POST event, and significant round-trip delay, which is clearly more costly than calculation of a hash. Third, the load may exceed legitimate traffic: in a service with one million users where the average user logs in once per day, a total of one billion guesses (one thousand guesses per account) is as many login requests as the legitimate population generates in 3 years. If legitimate users fail 5% or so of the time (due to typos, forgetting, etc.) the attacker will generate as many fail events as the legitimate population generates in 60 years.

Choosing a password to withstand an offline attack is thus much more difficult than choosing one to withstand an online attack. Yet the additional effort pays off only in the very restricted circumstances in which offline attacks can occur [15]. It makes little sense to focus on this risk when offline attacks are dwarfed by other vectors (such as malware).

3 Today’s over-constrained world

3.1 Password replacement schemes

Passwords offer compelling economic advantages over the alternatives, with the lowest start-up and incremental costs per user. Due largely to their status as the incumbent solution, they also have clear “deployability” advantages (such as backwards compatibility, interoperability, no migration costs). But it is not these factors alone that are responsible for their longevity; the “password replacement problem" is both under-specified and over-constrained [20, 6].

It is underspecified in that there is no universally agreed set of concrete requirements covering diverse environments, technology platforms, cultures and applications; for example, many authentication proposals become utterly unworkable on mobile devices with touchscreens, many Asian languages are now typed with constant graphical feedback that must be disabled during password entry, and many large websites must support both low-value forum accounts and important e-commerce or webmail accounts through a single system. It is simultaneously over-constrained, in that no single solution can be

[1] Freely available tables quickly reverse the MD5 hash of any alphanumeric password up to 10 characters. Using the passwords from RockYou (a site that had a major password leak) we can estimate this would cover 99.7% of users for low-value online accounts.

expected to address all requirements, ranging from financial to privacy protection. The list of usability, deployability and security requirements is simply too long (and rarely documented explicitly).

An in-depth review of 35 proposed password alternatives using a framework of 25 comparison criteria found no proposal beat passwords on all fronts. Passwords appear to be a Pareto equilibrium, requiring some desirable property X be given up to gain any new benefit Y, making passwords very difficult to replace.

Reviewing how categories of these password alternatives compare to regular passwords yields insight. Password managers–software that can remember and automatically type passwords for users—may improve security and usability in the common case, but are challenging to configure across all user agents. This problem also affects some graphical password schemes [4], while others offer insufficient security gains to overcome change-resisting inertia. Biometric schemes, besides their significant deployment hurdles, appear poorly suited to the unsupervised environment of Web authentication; fraudsters can just replay digital representations of fingerprints or iris patterns. Schemes using hardware tokens or mobile phones to generate one-time access codes may be promising, with significant security advantages, but ubiquitous adoption remains elusive due to a combination of usability issues and cost.

Federated authentication, or "single sign-on" protocols, in which users are authenticated by delegation to central identity providers, could significantly reduce several problems with passwords without completely eliminating them. Yet, besides introducing serious privacy issues, they have been unable to offer a business model sufficiently appealing to relying sites. The most successful deployment to date, Facebook Connect (a version of OAuth), incentivizes relying parties with user data, mandating a central role for Facebook as the sole identity provider, which does little for privacy.

With no clear winner satisfying all criteria, inertia becomes a substantial hurdle and the deck is stacked against technologies hoping to replace passwords entirely. A better choice is to prioritize competing requirements depending on organizational priorities and usage scenarios and aim for gradual adoption. Given their universal support as a base user-authentication mechanism, passwords are sensibly implemented first, offering the cheapest way to get things up and running when an idea is not yet proven and security is not yet critical, with no learning curve or interoperability hurdles. Low adoption costs also apply to users of new sites, who need low barriers when exploring new sites they are not sure they will return to. Financial websites are the rare exception, with offline capital and users whose accounts are all clearly valuable.

The list of challenges to would-be alternatives goes on. Improving security despite any decline in usability may mean losing potential new users (and sometimes existing users) to competitors. Some alternatives require server or client software modifications by various independent parties which is often a show-stopper; some others expect large numbers of users to change their existing habits or become trained to use new mechanisms; some are only partial solutions, i.e., address only a subset of security threats; some are even less user-friendly, though in new and different ways; and, as mentioned earlier, some are more costly and bring other deployment challenges (such as interoperability, compatibility with existing infrastructure, painful migration).

3.2 Advice to users

Users face a plethora of advice on passwords: use a different one for each account; change it often; use a mix of letters, punctuation, symbols and digits; make it at least eight characters long; avoid personal information (such as names or birthdays); and do not write them down. These suggestions collectively pose an unrealistic burden and are sometimes mutually incompatible; a person cannot be expected to memorize a different complex password for each of, say, 50 accounts, let alone change all of them on a rolling basis. Popular wisdom has summarized the password advice of the security experts as "Pick something you cannot remember and do not write it down."

Each bit of advice may be useful against a specific threat, motivating security professionals to offer them in an attempt to cover their bases and avoid blame for any potential security breaches regardless of the inconvenience imposed on users. This approach is predicted to lead to failure by the "compliance budget" model [3] in which the willingness of each user to comply with annoying security requirements is a finite, exhaustible resource that should be managed as carefully as any other budget. Indeed, websites (such as online stores), whose users are free to vote with their wallets, are much more careful about not exceeding their customers' compliance budget than sites such as universities whose users are "captive" [14].

Useful security advice requires a mature risk-management perspective and rough quantification of the risks and costs associated with each countermeasure. It also requires acknowledging that, with passwords as deployed today, users have little control over the most important countermeasures. In particular, running a personal computer free of malware may be the most important step, though it is challenging and is often ignored in favor of password advice, which is simpler to conceptualize but far less important. Likewise, good rate limiting and compromise detection on the server side are critical (see Sections 2.3 and 4), but users have no agency other than to patronize better-implemented sites.

Choosing extremely strong passwords, as is often advised, is of far more limited benefit; evidence that it reduces harm in practice is elusive. As noted earlier, password cracking is rarely a critical step in attacks. Hence making passwords strong enough to resist dedicated cracking attacks seems an effort poorly rewarded for all but the most critical Web accounts. For important accounts, password-selection advice should encourage passwords not easily guessed by acquaintances and sufficient for withstanding a reasonable amount of online guessing, perhaps one million guesses. About half of users are already above this bar [5], but discouraging simple dictionary passwords via advice, strength meters, and blacklists remains advisable to help out the others.

Advice to avoid reusing passwords is also common. While it is a good defense against cross-site password compromise, it is, for most users, incompatible with remembering passwords. Better advice is probably to avoid re-using passwords for *important* accounts and not to worry about the large number of accounts of little value to an attacker (or their owner).

Moreover, we consider the advice against writing passwords down to be outmoded for the Web. Stories involving "post-it notes on the monitor" usually refer to corporate passwords where users feel no personal stake in their security. Most users understand written-down passwords should be kept in a wallet or other safe location generally not available to others, even acquaintances. With this caveat, written passwords are a worthwhile trade-off if they encourage users to avoid very weak passwords. Password managers can be an even better trade-off, improving usability (no remembering, no typing) and allowing a different strong password for each account. However, they introduce a single point of failure, albeit perhaps no more vulnerable than webmail accounts already are due to the prevalence of email-based password reset.

4 A multi-dimensional future

We appear stuck between the intractable difficulty of replacing passwords and their ever-increasing insecurity and burden on users. Many researchers have predicted that the dam will burst soon and the industry will simply have to pay the necessary costs to replace passwords. However, these predictions have been made for over a decade. The key to understanding how large service providers manage, using what appears to be a "broken" technology, is that Web sites do not need perfection. The problem of compromised accounts is just one of many forms of abuse along with spam, phishing, click fraud, bots, fake accounts, scams and payment fraud. None of them has been completely defeated technologically, but all are managed effectively enough to keep things running.

In nearly every case, techniques that "solve" the problem technically have lost out to ones that manage them statistically; for example, despite many proposals to end spam, including cryptographic protocols to prevent domain spoofing, and micro-charges for each email message sent, most email providers have settled for approaches that classify mail based on known patterns of attacker behavior. These defenses are not free or easy to implement, with large Web operators often devoting significant resources towards keeping pace with abuse as it evolves. Yet, ultimately, this cost is typically far less than any approach requiring users to change behavior.

In the case of authentication, banks provide a ready example of living with imperfect technology.

Even though credit-card numbers are essentially static secrets, which users make no attempt to conceal from merchants, fraud is kept to acceptable levels by back-end classifiers. Technologies like "chip and PIN" have not been a magic bullet where it has been deployed [2]. Cards are still stolen, PINs can be guessed or observed, signature transactions still exist as a fallback, and online payments without a PIN, or "card not present" transactions, are still widespread.

Yet banks survive with a non-binary authentication model where all available information is considered for each transaction on a best-effort basis. Web authentication is converging on a similar model, with passwords persisting as an imperfect signal supplemented by many others.

4.1 Web authentication as classification

Behind the scenes, many large websites have already transitioned to a risk-based model for user authentication. This approach emerged by the early 2000s at online financial sites [41]. While an incorrect password means that access should be denied, a correct password is just one *signal* or *feature* that can be used by a *classifier* to determine whether or not the authentication attempt involves the genuine account owner.

The classifier can take advantage of many signals besides the password, including the user's IP address; geolocation; browser information, including cookies; the time of the login; how the password is typed; and what resources are being requested. Unlike passwords, these *implicit* signals are available with no extra effort from the user [21]. Mobile devices introduce many new signals from sensors that measure user interaction [11]. While none of these signals is unforgeable; each is useful. For example, geolocation can be faked by a determined adversary [28]; and browser fingerprinting techniques appear to be an endless arms race [29]. Nonetheless, both may be valuable in multi-dimensional solutions as the difficulty of forging all signals can be significant in practice; for example, by combining 120 such signals Google reported a 99.7% reduction in 2013 in the rate of accounts compromised by spammers [18].

Unlike traditional password authentication, the outcome is not binary but a real-valued estimated likelihood that the attempt is genuine. Generally these will to be discretized as users must be given access or not, and any classifier will inevitably make *false accept* and *false reject* errors. Sites will continue to develop their machine-learning techniques to reduce these errors, and may deploy new technology (such as two-factor authentication or origin-bound certificates [17]) to increase the number (and quality) of signals available.

Web authentication is by no means an easy domain for machine learning. The trade-off between false accepts and false rejects is difficult to get right. For financial sites, false accepts translate to fraud, but can usually be recovered from by reversing any fraudulent payments. However, for sites where false accepts result in disclosure of sensitive user data, the confidentiality violations can never be undone, making them potentially very costly. Meanwhile, false rejects annoy customers, who may switch to competitors.

Obtaining a large sample of *ground truth* to train the classifier is another challenge, as it is difficult to find examples of attacks that administrators do not yet know about. Financially motivated attackers are again likely the easiest to deal with, as their attacks typically need to be scalable, leading to a large volume of attacks and hence training data. Non-financially motivated attackers (such as ex-partners) may be more difficult to detect algorithmically, but users are far better positioned to deal with them in real life. Targeted attacks, including "advanced persistent threats" which are technically sophisticated and aimed at a single user account, are the most difficult challenge as attackers can tailor techniques to victims and leave relatively little signal available for classifiers.

4.2 New modes of operation

Authentication by classification enables fundamentally new methods of operation. Authentication can be a more flexible process, with additional information demanded as needed if the classifier's confidence is low or the user attempts a particularly sensitive operation, a process called *progressive* authentication [33]; for example, a site may ask users to confirm their identity by SMS or phone call if it notices suspicious concurrent activity on their account from geographically distant locations.

Multi-level authentication becomes possible, with users given limited access when the classifier's confidence is relatively low. In the UK, for example, some banks offer a read-only view of the account with just a password, but require a security token to transfer money out. Sites may also ask for less information, including not requiring a password to be entered, when they have reasonable confidence, from secondary signals, that the correct user is present. A form of this is already in place—where persistent session cookies once allowed password-less login for a predetermined duration the decision of when to re-check the password is now made by a classifier. A stronger version is *opportunistic* two-factor authentication, ensuring correct authentication when the second factor is present but enabling fallback security if the password is still correct and enough additional signals are presented [17].

The limit of this evolution is *continual authentication*. Instead of simply checking passwords at the entrance gate, the classifier can monitor the actions of users after letting them in and refine its decision based on these additional signals. Ultimately, continual authentication may mean the authentication process grows directly intertwined with other abuse-detection systems.

4.3 Changes to the user experience

As sites aim to make correct authentication decisions "magically" on the back-end through machine learning, most changes to the user experience should be positive. Common cases will be increasingly streamlined; users will be asked to type passwords (or take other explicit action) as rarely as possible. However, users also face potential downsides as systems grow increasingly opaque and difficult to understand.

First, users may see more requests for second factors (such as a one-time code over SMS) when the classifier's confidence is low. Users may also face more cases (such as when traveling or switching to a new machine) where they are unable to access their own account despite correctly remembering their password, akin to unexpected credit-card rejections while abroad. Increased rejections may increase the burden on "fallback" authentication, to which we still lack a satisfactory solution.

As authentication systems grows in complexity, their automated decisions may cause users increased confusion and distress. Users are less likely to buy in to any system that presents them with inconveniences they do not understand. Training users to respond with their credentials to asynchronous security challenges on alternative channels may also pave the way for novel phishing attacks. Even with careful user interface design, users may end up confused as to what the genuine ceremony [22] should be.

Another challenge is that better classifiers may break some access control practices on top of passwords which users have grown accustomed to. For example, users who share passwords with their spouses or their assistants may face difficulty if classifiers are able to (correctly) determine that another human is using their password, even though this is what the user intended.

Finally, typing passwords less often could in fact decrease their usability, as users are more likely to forget them if they go long periods between needing to type them.

4.4 Advantages of scale

Authentication may represent a classic example of the winner-take-all characteristics that appear elsewhere on the Web, since it offers benefits to scale in two different ways: First, large services are more likely to be accepted by relying parties as an identification service. Being accepted by more relying parties in turn encourages users to register accounts, further enhancing the attractiveness of these identity providers to relying parties. The second two-sided market (or positive feedback loop) is for user data. Large services with more user data can provide more accurate authentication. This attracts users to interact with these services more frequently, providing even more data for authentication. Developing and maintaining a complex machine-learning based approach to authentication requires a relatively large fixed technical cost and low marginal costs per user, further advantaging the largest identity providers.

One consequence of this consolidation is that, lacking access to the volumes of real-world data collected by larger service providers, independent researchers may be limited in their ability to contribute to several important research topics for which the limits of artificial datasets and mental extrapolation make empirically grounded research essential. Other areas of Web research (such as networking, which requires massive packet capture or search-engine research which requires huge numbers of user queries) have likewise become difficult for researchers with access to only public data sources.

There are also troubling privacy implications if relying parties require users to sign up with a large service that, in turn, requires a significant amount of personal information to perform authentication well. This information may be inherently sensitive (such as time and location of login activity) or difficult to change if leaked (such as behavioral biometrics like typing patterns). Many users already trust highly sensitive information to large online services, but authentication may be a motivating factor to collect more data, store it for longer and share it with more parties.

5 Conclusion

Passwords offer plenty of examples of divergence between theory and practice; estimates of strength, models of user behavior and password-composition policies that work well in theory generally remain unsupported by evidence of reduced harm in practice and have in some cases been directly contradicted by empirical observation. Yet large Web services appear to cope with insecure passwords, largely because shortcomings can be covered up with technological smarts in the back end. This is a crucial, if unheralded, evolution, driven largely by industry, which is experienced in data-driven engineering. Researchers who adapt their models and assumptions to reflect this trend will be in a stronger position to deliver relevant results. This evolution is still in its early stages and there are many important and interesting questions about the long-term results that have received little or no study to this point. There is also scope for even more radical rethinking of user authentication on the Web; clean-slate approaches may seem too risky for large companies but can be explored by more agile academic researchers. Tackling these novel challenges is important to ensure that published research is ahead of industry practice, rather than the other way around.

Acknowledgements

We thank Ross Anderson, Bill Cheswick, Richard Clayton, Lorrie Cranor, Arvind Narayanan, Avi Rubin, Max Spencer Ding Wang and Jeff Yan for insightful comments, though responsibility for the opinions herein remains ours. Joseph Bonneau is funded by a Secure Usability Fellowship from Simply Secure and the Open Technology Fund. Paul van Oorschot is funded by a Natural Sciences and Engineering Research Council of Canada Research Chair in Authentication & Computer Security and a Discovery Grant. Frank Stajano is partly supported by European Research Council grant 307224.

References

[1] A. Adams and M. Sasse. Users Are Not The Enemy. *Communications of the ACM*, 42(12):41–46, 1999.

[2] R. Anderson, M. Bond, and S. J. Murdoch. Chip and spin. *Computer Security Journal*, 22(2), 2006.

[3] A. Beautement, M. A. Sasse, and M. Wonham. The Compliance Budget: Managing Security Behaviour in Organisations. *NSPW*, 2008.

[4] R. Biddle, S. Chiasson, and P. C. van Oorschot. Graphical Passwords: Learning from the First Twelve Years. *ACM Computing Surveys*, 44(4), 2012.

[5] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. *IEEE Symposium on Security and Privacy*, 2012.

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

[7] J. Bonneau and S. Preibusch. The password thicket: technical and market failures in human authentication on the web. *Workshop on the Economics of Information Security (WEIS)*, June 2010.

[8] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, , W. T. Polk, S. Gupta, and E. A. Nabbus. Electronic Authentication Guideline. *NIST S.P. 800-63-2*, 2013.

[9] W. R. Cheswick. Rethinking passwords. *Communications of the ACM*, 56(2):40–44, 2013.

[10] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The Tangled Web of Password Reuse. *NDSS*, 2014.

[11] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and I know it's you!: Implicit authentication based on touch screen patterns. *ACM CHI*, 2012.

[12] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley. Does My Password Go Up to Eleven?: The Impact of Password Meters on Password Selection. *ACM CHI*, 2013.

[13] D. Florêncio and C. Herley. A large-scale study of web password habits. *16th International Conference on the World Wide Web (WWW)*, 2007.

[14] D. Florêncio and C. Herley. Where Do Security Policies Come From? *ACM SSOUPS*, 2010.

[15] D. Florêncio, C. Herley, and P. C. van Oorschot. An Administrator's Guide to Internet Password Research. *Usenix LISA*, November 2014.

[16] S. Garera, N. Provos, M. Chew, and A. D. Rubin. A Framework for Detection and Measurement of Phishing Attacks. *ACM Workshop on Recurring Malcode (WORM)*, 2007.

[17] E. Grosse and M. Upadhyay. Authentication at Scale. *IEEE Security & Privacy Magazine*, 11:15–22, 2013.

[18] M. Hearn. An update on our war against account hijackers. Google Security Team Blog, Feb 2013.

[19] C. Herley. So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users. In *NSPW*. ACM, 2009.

[20] C. Herley and P. C. van Oorschot. A Research Agenda Acknowledging the Persistence of Passwords. *IEEE Security & Privacy*, 10(1):28–36, 2012.

[21] M. Jakobsson, E. Shi, P. Golle, and R. Chow. Implicit authentication for mobile devices. *USENIX Conference on Hot Topics in Security (HotSec)*, 2009.

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

[23] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. *IEEE Symposium on Security and Privacy*, 2012.

[24] D. Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. In *2nd USENIX Security Workshop*, 1990.

[25] J. L. Massey. Guessing and Entropy. In *Proceedings of the 1994 IEEE International Symposium on Information Theory*, 1994.

[26] R. Morris and K. Thompson. Password security: a case history. *Commun. ACM*, 22(11):594–597, 1979.

[27] D. M'Raihi, S. Machani, M. Pei, and J. Rydell. TOTP: Time-Based One-Time Password Algorithm. RFC 6238, IETF, May 2011.

[28] J. A. Muir and P. C. van Oorschot. Internet geolocation: Evasion and counterevasion. *ACM Computer Surveys*, 42(1), 2009.

[29] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. *IEEE Symp. Security & Privacy*, 2013.

[30] D. A. Norman. THE WAY I SEE IT: When Security Gets in the Way. *interactions*, 16(6), November 2009.

[31] P. Oechslin. Making a Faster Cryptanalytic Time-Memory Trade-Off. In *Advances in Cryptology (CRYPTO)*, August 2003.

[32] J. O. Pliam. On the Incomparability of Entropy and Marginal Guesswork in Brute-Force Attacks. In *INDOCRYPT '00: The 1^{st} International Conference on Cryptology in India*, 2000.

[33] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. Progressive authentication: deciding when to authenticate on mobile phones. *USENIX Security*, 2012.

[34] S. Schechter, C. Herley, and M. Mitzenmacher. Popularity is Everything: A new approach to protecting passwords from statistical-guessing attacks,. *USENIX HotSec*, 2010.

[35] E. Spafford. Observations on Reusable Password Choices. *USENIX Security Workshop*, 1992.

[36] F. Stajano. Pico: No more passwords! *Security Protocols Workshop (SPW)*, 7114, 2011.

[37] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor. How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation. *USENIX Security*, 2012.

[38] U.S. Department of Defense. Password management guideline. Technical Report CSC-STD-002-85, 1985.

[39] U.S. National Institute of Standards and Technology. Password Usage. Federal Information Processing Standards Publication 112. May 1985.

[40] M. Weir, S. Aggarwal, M. Collins, and H. Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. *ACM CCS*, 2010.

[41] G. D. Williamson. Enhanced authentication in online banking. *J. of Econ. Crime Management*, 4(2), 2006.

[42] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password Memorability and Security: Empirical Results. *IEEE Security & Privacy*, 2(5):25–31, 2004.

[43] Y. Zhang, F. Monrose, and M. Reiter. The Security of Modern Password Expiration: An Algorithmic Framework and Empirical Analysis. *ACM CCS*, 2010.