A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords

Alexandra Nisenoff [†*], Maximilian Golla[†‡], Miranda Wei[†⋆], Juliette Hainline[†], Hayley Szymanek[†], Annika Braun[†], Annika Hildebrandt[†], Blair Christensen[†], David Langenberg[†], Blase Ur[†]
† University of Chicago, ∗ Carnegie Mellon University,
‡ Max Planck Institute for Security and Privacy, ⋆ University of Washington

Abstract

Credential-guessing attacks often exploit passwords that were reused across a user's online accounts. To learn how organizations can better protect users, we retrospectively analyzed our university's vulnerability to credential-guessing attacks across twenty years. Given a list of university usernames, we searched for matches in both data breaches from hundreds of websites and a dozen large compilations of breaches. After cracking hashed passwords and tweaking guesses, we successfully guessed passwords for 32.0% of accounts matched to a university email address in a data breach, as well as 6.5% of accounts where the username (but not necessarily the domain) matched. Many of these accounts remained vulnerable for years after the breached data was leaked, and passwords found verbatim in breaches were nearly four times as likely to have been exploited (i.e., suspicious account activity was observed) than tweaked guesses. Over 70 different data breaches and various username-matching strategies bootstrapped correct guesses. In surveys of 40 users whose passwords we guessed, many users were unaware of the risks to their university account or that their credentials had been breached. This analysis of password reuse at our university provides pragmatic advice for organizations to protect accounts.

1 Introduction

Despite their disadvantages, passwords remain widely used for authentication [7]. Organizations must protect against large-scale attacks on users' passwords. An adversary may leverage **reused passwords**—when the same individual picks similar or identical passwords for different services [10, 80] to cope with having to remember numerous passwords [16]. If any one of these services suffers a data breach, attackers typically try to log into another service with the same email address alongside a password that is either the same as the leaked password, or tweaked in small ways. Such credential-stuffing attacks are this paper's focus. Additionally, attackers may guess the **common passwords** most frequently chosen across all users [6], which we also study for contrast.

The ability to conduct attacks that exploit reused password has increased as hundreds of websites have had their password databases stolen and leaked over the last decade [34]. We term the breach of a single service an **individual service breach**. In recent years, hackers have also packaged credentials from many different services into **breach compilations** containing hundreds of millions or even billions of credentials [24].

To protect an organization against attacks exploiting common passwords, system administrators can institute straightforward blocklists [25, 70]. Protecting an organization from reused passwords, however, is far more complex. A vulnerable password is specific to one user based on their credentials on other sites at any past or future time. Furthermore, prospective attackers often have far more information than system administrators. Attackers may know about a successful breach that system administrators may not hear about for years, or ever. Further, attackers may pool resources to crack hashes and reveal the plaintext needed for an attack, while the system administrator may be left only with uncracked hashes [9].

In recent years, researchers and practitioners have developed compromised-credential-checking tools to try to defend users. For instance, Chrome [73], Firefox [55], and Safari [12] notify users if their passwords appear in a data breach. The Have I Been Pwned (**HIBP**) service [32], itself integrated with 1Password [13], enables users to check for their appearance in a data breach. Supporting these efforts, academic work has proposed protocols that underpin compromised-credential-checking tools [40, 43, 44, 59, 82, 83] and sought to improve the usability of data breach notifications [22, 31, 53, 79, 90, 92].

Despite prior work, many questions remain for system administrators trying to protect their organizations from attacks exploiting reused passwords. For what amount of time are accounts vulnerable? Out of hundreds of data breaches, how important is it to account for them all? Should defenders devote resources to trying to crack hashes to protect users? Is it sufficient to look for matching email addresses, or should they also search for matching usernames? How often do attackers appear to have exploited reused passwords, and what factors make them more likely to have done so?

We answer these questions, and more, through a twenty-year retrospective analysis of our university's vulnerability to password-guessing attacks and companion survey of affected users. This analysis was possible because our university's password-composition policy prohibits a user from ever returning to one of their previously used passwords, which requires maintaining a **password history database** (a time-stamped log of historical password hashes) and comparing against it whenever a user submits a new password. When we learned about this unique data source, we realized how valuable it could be for gaining insight into the longitudinal aspects of reused and compromised credentials. Through a collaboration between academic researchers and both the IT Security and Identity Management teams at our university, this project aimed not just to create generalizable knowledge about password reuse and compromised credential checking, but also to directly improve our university's security by forcing password resets for any user whose password we guessed.

We carefully designed the study, which was approved by our institution's IRB, to minimize risk to accountholders at our university and to reduce their own vulnerability. Starting with a list of roughly 225,000 usernames of accounts held by faculty, staff, and students at our university over the past twenty years, the academic researchers in our team searched over 450 individual service breaches and 12 breach compilations for credentials either associated with an email address at our university or sharing a username—either in isolation or as part of an email address at a different domain (e.g., `bob@uchicago.edu` vs. `bob` vs. `bob@gmail.com`). When we found hashes, rather than plaintext credentials, we attempted to crack them. We then used four state-of-the-art methods [10,58,67,80] to tweak credentials (e.g., `monkey1` → `Monkey1!`). We then sent guesses (usernames and passwords) alongside metadata about how each guess was generated to the IT Security team, who compared these guesses to the password history database. We also provided common passwords to guess for all accounts. For correct guesses, the IT Security team returned pseudonymous metadata (without usernames and passwords) augmented with additional metadata (e.g., when the password was created). They also forced password resets for users whose current password was guessed.

Exploiting password reuse, we successfully guessed passwords for 32.0% of accounts matched to a university email address in a data breach and 6.5% of accounts with any potential username or email match. For 35.5% of accounts for which we correctly guessed any password, we guessed the user's current password. Common password guesses were significantly less successful, underscoring the far greater risk posed by attacks leveraging reused passwords even if (as we did) common passwords are customized for the attacked service. Although 71 individual service breaches and 12 breach compilations bootstrapped at least one correct guess, the breaches of LinkedIn, Chegg, LiveJournal, Dropbox, and MySpace each bootstrapped over 500 correct guesses. Credentials from LinkedIn were particularly effective at guessing employees' passwords, and credentials from Chegg (a homework help site) at guessing students' passwords.

Many accounts remained vulnerable for years. Five years after a given breach was made public, roughly half of affected accounts remained vulnerable. While the peak vulnerability to an individual service breach was often around when the breach occurred (and before it was made public), breach compilations were typically made public a few years after peak vulnerability. The university changing the minimum length of newly created passwords from 8 to 12 characters in 2015 was a key inflection point in reducing vulnerability.

Though 54.7% of correct guesses were based on **verbatim reuse** (exactly matching the breached password), the rest required password tweaking using four previously published methods [10,58,67,80]. Toggling the case of the first character and appending either "!" or "1" were the most successful strategies. While a recent deep-learning-based approach [58] produced the best ordered list of transformations "out of the box," earlier heuristics-based methods [10,80] may have been more successful had their guesses been optimally ordered.

We also studied whether attackers seem to have exploited these vulnerabilities. When our IT Security team detects suspicious activity on an account, it locks the account and forces a password reset, logging these actions. On 29 separate days over the last eight years, the IT Security office observed suspicious activity on ten or more accounts whose passwords we guessed. Passwords found verbatim in breaches were nearly four times as likely to have been exploited, whereas passwords found in plaintext (versus hashed) were only somewhat more likely to have been exploited. Surprisingly, most credentials we guessed did not seem to have been exploited previously by attackers, underscoring organizations' latent risk.

Finally, we surveyed 40 university affiliates whose passwords we guessed to understand their experiences and knowledge. Confirming prior work [49], most respondents were unaware of the risks to their university account. Several were not even aware they had an account on the breached site.

While a few prior papers [10,58,65,72,80] measured some aspects of password reuse, our retrospective approach enabled numerous novel findings and lessons for organizations. We found that an organization's vulnerability to password-reuse-based attacks can vary greatly over time. Not considering the long tail of available data breaches or more permissive (imprecise) strategies for matching accounts can lead to an incomplete view of vulnerability. A careful reordering of heuristic methods for tweaking passwords might outperform deep-learning methods. Vulnerable credentials can remain in use for a long time even if an organization follows best practices. The exploitation of accounts at our university mostly did not leverage password tweaking or imprecise account matching. Many vulnerable passwords were created at our university before the corresponding data breach, posing problems for credential checking at the time of password creation.

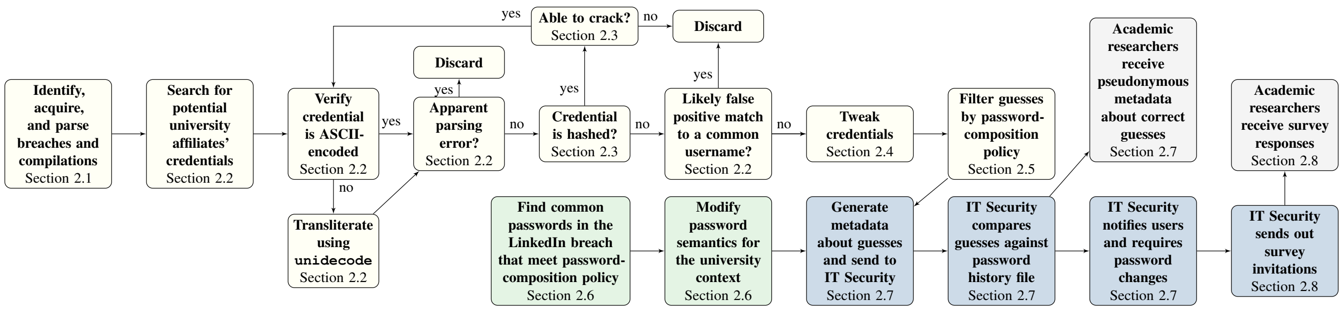

Figure 1: Overview of our study procedure.

2 Methods

Here, we detail how academic researchers and our university's IT Security team (**ITS**) collaborated both to answer research questions and to reduce the university's vulnerability to attacks while minimizing risk to users. We relied on the aforementioned password history database, a time-stamped log of the hashes of every password used by university affiliates since late 2002. Figure 1 summarizes our approach.

Accounts at our university are single-sign-on accounts that provide access to a wide range of services, including email, payslips, academic records, and systems needed for staff, faculty, and students to do their work. When a student graduates or employee leaves, their account remains active with limited access (e.g., forwarding email and accessing tax / academic records). The university recently required current faculty, staff, and students to use Duo two-factor authentication (2FA).

2.1 Sources of Leaked Passwords

We bootstrapped credential guesses by searching over 450 individual service breaches[1] and 12 large breach compilations for credentials potentially associated with a university affiliate's other online accounts. We selected sources in several ways. Initially, members of our team scanned through HIBP's list of "Pwned Websites" [34] to identify sources likely to include credentials from university affiliates based on the size of the breach and service's regional focus. We required that sources include passwords as well as either email addresses or usernames. The selected sources included both individual service breaches (e.g., Neopets) and breach compilations (e.g., Collection #1) containing credentials from many different sources grouped together. We augmented this list with commonly discussed sources not explicitly listed on HIBP (e.g., Collections #2–5). We obtained data from public websites and from personal contacts in the password cracking community. In doing so, we did not sign up for any private leak forums, pay any money, redistribute any data, or use any method of downloading that would facilitate others obtaining the data.

[1]The provenance and identity of files leaked publicly often cannot be verified. Some files may be the spoils of phishing attacks (rather than stolen password databases), be mislabeled, or mix credentials from multiple sources.

Following matching (Section 2.2) and filtering (Section 2.5), we generated at least one guess based on 267 individual service breaches and all 12 breach compilations. These breaches were made public between 2008 and 2020. Our analysis of 190 additional individual service breaches did not yield any compliant guesses. The abbreviated Table 7 later in the body of the paper and full table in our extended version [56] detail the individual service breaches that bootstrapped at least one correct guess (i.e., match in the password history database). Table 8 does the same for breach compilations.

2.2 User Matching & Data Sanitization

We used a list of 227,976 usernames in our university's password history database as the starting point for three ways of identifying potential matches in individual breaches and compilations. The first, an **exact email match**, was when a password or hash in the breach or compilation was associated with a university email address (`username@uchicago.edu` or `username@subdomain.uchicago.edu`). We excluded email addresses whose username did not appear in the password history database.[2] The second, a **similar email match**, was when a username from the password history database matched the username for a non-university email address (e.g., `username@gmail.com`) associated with the leaked credential. Third, a **username match** was when the username (for services that had standalone usernames) associated with the leaked credential exactly matched the university username.

Most prior work only considered exact email matches, not similar email matches or username matches. While we expected these strategies to result in a large fraction of false positives in matching, we wanted to understand to what extent system administrators should account for imprecise matching strategies in performing compromised credential checking.

We refer to passwords found in breaches that are potentially associated with a university affiliate as **leaked passwords**. To focus on likely instances of password reuse relative to our university's historical password-composition policies (Section 2.5), we discarded leaked passwords shorter than

[2]The university permits affiliates to create aliases (alternate email addresses), but the alias cannot be used to log into any university resources.

six characters. After filtering, we obtained 35,040,844 possible credentials (including uncracked hashes) associated with 189,984 of the 227,976 users in the password history database.

We performed further sanitization. Our university only allows ASCII characters in passwords, so we used Python's `unidecode` package to convert non-ASCII characters. We used heuristics to identify and discard leaked passwords that likely resulted from parsing errors by the hackers who leaked the data (e.g., IP addresses, email addresses, passwords containing HTML). Similar email matches and username matches on common usernames (e.g., `bob`) were likely to produce huge numbers of false positives (i.e., not be the university affiliate) and make deep-learning-based credential tweaking intractable. We thus discarded such matches with 100+ unique leaked credentials, but retained all exact email matches.

2.3 Cracking Hashes

While some services that suffered data breaches ill-advisedly stored passwords in plaintext, most hashed passwords. Thus, individual service breaches and breach compilations sometimes contained only plaintext passwords, sometimes contained only hashes, and sometimes contained a mix (as a result of the attackers or security community cracking hashes).

For matches containing hashes, we followed a best-effort approach to obtain the plaintext. We simulated an invested attacker with moderate cloud resources [4, 15, 17, 76]. As in prior work [72], we identified likely hashes by looking for fixed-length strings consisting of only hexadecimal characters. Members of our team with substantial experience in password cracking attempted to crack hashes using a combination of dictionaries and mangling rules (Hashcat's `best64`, `OneRuleToRuleThemAll`, and `dive` sets), as well as mask attacks (selective brute-forcing) for fast hash functions like MD5 and SHA-1. Beyond using large, untargeted dictionaries like `Hashes.org Founds` and `rockyou2021.txt`, we also created our own that included all plaintext leaked passwords across our sources. For slow hash functions like bcrypt, we only tested the one million most common passwords [52].

After searching through lists of already cracked hashes published online or on sites like hashes.org, approximately 2 million hashes without publicly available plaintext equivalents remained. We spent one week cracking. We recovered plaintext equivalents for 32% of the remaining hashes. While we were able to recover 57% of fast hashes like MD5 and SHA-1, we only cracked 11% of slow hashes like bcrypt. While this number may seem low, hashed credentials for which a plaintext equivalent is not public are those that others in the cracking community have themselves likely struggled to crack.

2.4 Credential Tweaking

Prior work has found that users often tweak passwords, or modify them in small ways, when reusing them across services [10]. Some studies have proposed algorithms for tweaking passwords. Both to support our measurements and to compare prior methods in our own context, we tweaked the leaked passwords we identified using three methods from prior academic papers, as well as a simple mangling-rule-based approach. Specifically, we tested heuristics-based methods from Das et al. [10] and Wang et al. [80], as well as the `pass2path` deep learning model from Pal et al. [58]. Because Das et al. [10] and Wang et al. [80] did not open-source their code, we re-implemented the methods described in their papers, asking for clarifications from the original authors over email. Pal et al. [58] shared their pass2path code with us. Due to computational limitations, we configured pass2path to generate only up to 150 transformations per leaked password.

Not every transformation attempt will modify a given password. For instance, replacing "e" with "3" results in no change for a password without an "e." Furthermore, we discarded transformations that did not comply with any of our university's password-composition policies (see Section 2.5). In the end, per leaked password, the approaches generated a mean of 134.3 (Das et al.), 363.6 (Wang et al.), and 59.5 (Pal et al.) unique guesses beyond the original that complied with a password-composition policy. These means are substantially smaller than the number of tweaks attempted (e.g., 59.5 vs. 150). As an additional point of comparison, we evaluated the Hashcat mangling rules optimized in the Best64 Challenge [67]. While not explicitly designed for credential tweaking, `best64.rule` is a de facto standard rule set shipped with software like Hashcat. It currently consists of 77 unique rules. It generated a mean of 27.4 unique and policy-compliant guesses per password beyond the original. Tweaked passwords were generated by processing all leaked passwords per user at a time. In our metadata, we merged guesses generated multiple times by either a single method or multiple methods.

Table 1: Key password-composition policy characteristics.

Policy	Length	Character Classes
Password (Jan 2015 – Present)	12 – 19	3+
Password (Apr 2010 – Jan 2015)	8 – 16	3+
Password (Prior to Apr 2010)	8 – 16	2+
Passphrase (Jan 2016 – Present)	18 – 32	1+
Passphrase (Aug 2014 – Jan 2016)	18 – 50	1+

2.5 Filtering by Password-Composition Policy

As summarized in Table 1, our university's current password-composition policy is that users may either create a password (12–19 characters with 3+ character classes) or a passphrase (18–32 characters with no character-class requirement). The policy has other facets we did not consider in generating guesses; see our extended version [56] for details. Most passwords do not expire; medical center staff are exceptions.

Table 2: A summary of the number of *leaked passwords* (appearing in individual service breaches or breach compilations) and the number of eventual *password guesses* (including tweaks) that complied with our university's password-composition policies.

	Leaked Passwords				All Password Guesses (Leaked + Tweaked)			
			Exact Email Match				Exact Email Match	
Policy	**# Passwords**	**# Users**	**# Passwords**	**# Users**	**# Passwords**	**# Users**	**# Passwords**	**# Users**
Password (Jan 2015 – Present)	65,254	38,865	736	688	286,081,420	128,557	2,118,287	5,472
Password (Apr 2010 – Jan 2015)	333,197	95,191	3,550	3,304	1,017,849,564	154,120	6,813,861	13,752
Password (Prior to Apr 2010)	1,415,055	139,039	10,493	9,056	1,523,723,163	156,611	9,660,165	14,322
Passphrase (Jan 2016 – Present)	22,111	15,975	167	139	26,655,433	81,373	432,189	1,550
Passphrase (Aug 2014 – Jan 2016)	24,555	17,330	169	140	27,954,255	84,027	442,040	1,680
Non-compliant	1,663,284	140,091	7,524	6,736	–	–	–	–
Total	3,104,557	156,618	18,205	14,328	1,562,510,968	156,618	10,265,787	14,328

There have been a few key changes over time that applied to newly created passwords. As such, existing passwords did not have to be changed when the policy changed. The minimum length required for passwords was increased to the current 12 characters from the previous 8 characters in January 2015. The minimum number of character classes was increased to the current 3+ from 2+ in April 2010. Beginning in August 2014, users could avoid character class requirements altogether by creating a passphrase (18+ characters).

While these requirements are more strict than many consumer-facing websites, policies requiring multiple character classes and relatively long passwords are common for organizations [19]. Thus, we expect our results to generalize most directly to other organizations, especially universities. In fact, our university's 2002-2015 password policy was the most commonly observed policy in a survey of organizations [19].

We use the term **password guess** to refer to either a candidate leaked password found verbatim in a breach (or compilation) or a candidate tweaked version of that password that complies with at least one of these composition policies. Any candidate that did not comply with any policy was discarded.

Following this filtering step, we had a total of 3,104,557 password guesses associated with 156,618 users. There was a median of 9 leaked passwords per user, and a mean of 19.8. Table 2 summarizes these password guesses and their compliance with the university's password-composition policies.

2.6 Choosing Common Passwords

To understand how an organization's exposure to password reuse compares to its exposure to common passwords, we also guessed common passwords for every user. These guesses were the most frequent (those that appeared at least ten times) in the individual service breach of LinkedIn, whose passwords have been studied in many other papers [5,20,27,29,36,45,60, 77,78]. The LinkedIn breach was a suitable source for multiple reasons: i) LinkedIn's focus on professional networking matches our organizational context; ii) it is a relatively large breach; iii) the vast majority of its hashes have already been cracked; and iv) its characteristics have been well-studied.

Table 3: Compliance of **common password guesses**.

Policy	**# Frequently Found in LinkedIn**	**# Guesses After Modification**
Password (Jan 2015 – Present)	377	2,377
Password (Apr 2010 – Jan 2015)	838	3,092
Password (Prior to Apr 2010)	1,219	3,621
Passphrase (Jan 2016 – Present)	121	130
Passphrase (Aug 2014 – Jan 2016)	121	130
Total	1,340	3,751

Using a single data breach, rather than aggregating across breaches, avoids issues of how to weight password frequencies from breaches of vastly different sizes from contexts, languages, populations, and password-composition policies that often differ from our university's. Because passwords sometimes relate semantically to the website for which they were originally created [85], we modified common password guesses related to LinkedIn itself (e.g., `LinkedIn123`) to instead reference our university (e.g., `UChicago123`), which was again possible due to our use of a single data breach. Specifically, we replaced substrings like "LinkedIn", "linked", and "link" with comparable strings related to our university. Since LinkedIn was breached in 2012, many passwords referenced years around then. For every password containing a number between 2002 and 2025, we replaced that number with all numbers between 2002 and 2025. Table 3 summarizes these common password guesses.

2.7 Generating Metadata and Testing Guesses

Alongside each password guess, the academic researchers included metadata about how that guess was generated. When returning data to the academic researchers, ITS kept the metadata, but removed usernames and passwords. This metadata included the breach(es) or compilation(s) in which we found the leaked password bootstrapping the guess, how the guess was tweaked (if at all), and whether the leaked password was hashed. ITS added metadata, such as the dates when the pass-

Table 4: Metadata we generated and collected about each password guess.

Category of Data	Source of Data	Reuse Guesses	Common Password Guesses
Username	Academic researchers	●	●
Password guess	Academic researchers	●	●
Individual service breaches and/or breach compilations in which the leaked password appeared	Academic researchers	●	○
Matching strategy used for the username (exact email, similar email, username)	Academic researchers	●	○
Whether the leaked password was found as a **hash or in plain text** in data breaches, as well as the hash format (if applicable)	Academic researchers	●	○
The candidate password's **compliance** with the University's password or passphrase policies	Academic researchers	●	●
Whether the password guess was leaked verbatim or **transformed**, including the transformations that generated it (if applicable)	Academic researchers	●	○
Whether the leaked password contained only **ASCII** characters; if not, it was converted using Python's `unidecode` package	Academic researchers	●	○
The **length** of the leaked password(s) and resultant password guess after transformations	Academic researchers	●	●
The **character classes** present in the password guess	Academic researchers	●	●
The **approximate strength** of the password guess, specifically the $\log_{10}$ of the number of guesses to crack it as estimated by zxcvbn [87]	Academic researchers	●	●
If the guess would have been in the top 50, 100, or 1000 guesses for each password-composition policy	Academic researchers	○	●
If the guess of a common password was created by modifying the password to be related to the university	Academic researchers	○	●
If the guess of a common password was created by modifying years that appeared in the original password	Academic researchers	○	●
A **randomized ID** for each user. A single ITS employee had the crosswalk mapping randomized IDs to usernames	IT Security Team	●	●
The **initial creation date** of the password	IT Security Team	●	●
Whether the password was: (a) currently valid at the time we provided ITS with this information (b) not currently valid, but previously valid (and on what date the password was changed and thus no longer valid)	IT Security Team	●	●
If the password was created as a result of: (a) a password reset that ITS compelled for security reasons (b) a user-initiated password change	IT Security Team	●	●
If the user's previous password stopped being valid as a result of: (a) a password reset that ITS compelled for security reasons (b) a user-initiated password change	IT Security Team	●	●
The user's **current affiliation** with the University (e. g., student, faculty, alumni)	IT Security Team	●	●
If that user has **2FA currently enabled** for their account	IT Security Team	●	●
If the account is **provisioned**, meaning it has not been disabled; in the past, accounts were disabled if an employee left the university	IT Security Team	●	●
If the user has ever been forced to reset a password due to a **security incident**, and the date(s) those occurred (if applicable)	IT Security Team	●	●

word was created and changed (or whether it remained active), whether that password change was mandated due to suspicious account activity, and the user's current university affiliation. Table 4 presents the full list of metadata.

Once all password guesses had been generated, the academic researchers GPG-encrypted them and transferred them to ITS. A single research contact at ITS checked the guesses against the password history database in July 2022. We term any password guess that matched a username and password a **correct guess**. A correct guess could be either **currently valid**—that user's current password—or **previously valid**.

To reduce our university's vulnerability, ITS forced affiliates whose current password was guessed to choose a new password. After a 14-day grace period, accounts with unchanged passwords were locked and could be reset through the university's help desk. Additionally, ITS sent courtesy notifications to users whose current password was not guessed, but whose recent password (used in the past three years) was guessed. In all cases, notifications described the research, explained the dangers of password reuse, and gave participants the opportunity to withdraw their data from the research.

2.8 Survey of Impacted Users

To understand the experiences and attitudes of university affiliates who had reused their password, we conducted a survey. Our extended version [56] includes the survey instrument.

The ITS research contact emailed a survey invitation to a sample of 1,495 university affiliates whose current or recent (within the last three years) password we had guessed correctly. We preferentially sampled users who were current students or employees whose current password we had guessed. After finishing the survey, respondents received a \$10 Amazon gift voucher forwarded by the ITS research contact.

The survey began with a consent form that clarified that ITS could not access survey responses and the academic researchers would not know their identity. We then asked multiple-choice and open-ended questions about respondents' security practices and experiences with their university account. Next, we showed respondents details about the breach(es) and compilation(s) that enabled us to guess their password. While the original notification emails mentioned in general that data breaches were used, this was the first time they were shown the specific breaches. We queried their reaction to this information and knowledge of the breach(es). We finished by soliciting their perceptions of credential checking.

We received 40 survey responses. Among respondents, 30% were currently affiliated with the university. For 68% of respondents, we had guessed their current password, forcing a reset. The leaked password bootstrapping our guesses was found only in an individual service breach (30% of respondents), only in a breach compilation (48%), or in both (23%). Only one participant saw more than one individual service breach. The mean number of breach compilations was five.

2.9 Ethics

Given the sensitivity of passwords and account security, our team carefully designed this research protocol collaboratively with numerous stakeholders at our university over nearly five years. Properly handling user data and minimizing risk were primary concerns. Below, we discuss key safeguards.

IRB: We designed our protocol through many consultations with the prior and current directors of our university's IRB. Our IRB formally approved our protocol. The ITS team contacts also completed human-subjects protection training.

University Stakeholders: We refined our protocol through discussions with IT Leadership (including the CIO), the provost's office, the university's communications team, the university's general counsel, and the alumni association.

Informed Consent: Because notifying all university affiliates, most of whose passwords we expected not to guess, would burden them, our IRB granted our measurement study a waiver of informed consent. However, all users whose current or recent password was guessed were notified and given the opportunity to withdraw their data from the research, though they would still be required to change their password if applicable. Based on multi-stakeholder discussions, we decided not to inform users if none of the passwords we guessed were active in the last three years to avoid causing unneeded worry.

Password Reset: ITS forced any users whose current password was guessed to choose a new password, even if exploitation was not exceedingly likely (e.g., a cracked bcrypt hash tweaked using a rare strategy). To minimize the burden on users absent observed account compromise, we set a 14-day window for the password change, with regular reminders. We also timed this process to avoid stressful times (e.g., exams).

Education: The notifications sent to users reflected best practices for password-reuse notifications [22]. They included relevant information about the required reset and why password reuse is risky. The notifications included contact information for ITS, the IRB, and the principal investigator. They also linked to a webpage with password-security tips.

Compartmentalized Data Access: We minimized the access any team member had to the data collected. Some breaches include data beyond credentials. Only the academic researchers worked with these files, removing all data beyond the username and password. A single ITS employee accessed the password guesses and maintained the crosswalk between randomized IDs and actual usernames. The academic researchers never learned the usernames or passwords of correct guesses, only pseudonymous metadata. Furthermore, only two academic researchers had access to this metadata. The ITS team has access to the password history database as part of their regular job duties, adding no additional risk.

Preventing Re-identification: We intentionally balanced the richness of possible metadata with its risks. For instance, we calculated binned, inexact values for several types of metadata (e.g., password strength). All members of the team agreed not to make any attempts to re-identify any users. Furthermore, we only report aggregate statistics on the metadata.

No Redistribution or Payment: While obtaining individual service breaches and breach compilations, we did not sign up for any forums, pay any money, or redistribute the sources.

Survey: IT Services performed all recruitment and communication with respondents. The survey was conducted remotely, and only the academic researchers could access survey responses. If the credentials were from a sensitive source (e.g., an adult website), we would not display the source in the survey. The survey included "prefer not to answer" options. Upon completion, we provided tips for protecting accounts.

Nonetheless, the ethics of studying password data leaks are the subject of ongoing discussions [11, 35]. Prior work discussed harms and benefits [71] and studied how users feel about the use of this data in different contexts [37].

2.10 Limitations

As with any study, ours has limitations. While we aimed to simulate techniques used by attackers, our methods likely overestimate their capabilities in some ways, yet underestimate them in others. We started with a list of all valid usernames, whereas an attacker would need to compile their own (imperfect) list from the web or university directory. In addition, we did not need to worry about a large number of incorrect guesses triggering an alarm and thus made hundreds or thousands of guesses for some accounts. An attacker would need to spread guesses over time, accounts, and IP addresses.

Our handling of hashes likely contributed to both overestimates and underestimates. While we successfully cracked nearly a third of the hashes we found, enabling guesses low-resourced attackers could not make, well-connected and well-resourced attackers likely have access to additional breaches and cracking hardware. Attackers may also use entirely different attack strategies and cracking techniques.

The scope of our data also had limitations. Users engaging in password reuse will not appear in our dataset if none of the other services for which they use similar credentials have yet been breached. While our metadata includes users' current affiliations, we cannot recover historical affiliations at the time a password was created. Our data about which accounts were exploited was based on the ITS team's heuristics for suspicious activity, likely missing some account compromises. Finally, our university's accounts may have varying levels of importance to individuals, impacting the passwords selected.

Survey responses were limited by both participants' memory and recollections about their past actions, as well as their willingness to disclose information on topics that they may have found sensitive. Our sample was relatively small, further limiting the conclusions we can draw.

Our study of passwords at a university is more likely to generalize to other universities and organizations than to consumer-facing websites. Password-composition policies at organizations are more stringent than for other websites [19, 42]. Universities may be less inclined to delete accounts for inactive users, skewing vulnerability windows. Further, users differ in the importance they place on their university accounts, particularly once they leave the university (even though the accounts still contain sensitive information).

Table 5: Summary of correct guesses.

		Reused Passwords	Common Passwords
# currently valid passwords		3,618	696
% of users with any guesses made		2.3%	0.3%
Total # of passwords		12,247	1,979
# of unique users		10,186	1,705
% of users with any guesses made		6.5%	0.7%
Years password active:	Median	6.2	1.8
	IQR	1.4 - 12.0	0.2 - 8.1

3 Results

We **correctly guessed 14,161 passwords** contained in our university's password history database. **Reused passwords were a far greater vulnerability than common passwords.** As detailed in Table 5, 12,247 of these correct guesses exploited reused passwords affecting 10,186 users. This corresponds to 4.5% of all users in the password history database and 6.5% of the users for whom we made at least one password reuse based guess, which required at least one leaked password. This percentage was far higher for users with an exact email match (i.e., associated with a `uchicago.edu` email address). **We correctly guessed at least one password for 32.0% of the 14,328 users with an exact email match**. Of these guesses, 3,618 matched a user's current password.

Meanwhile, while only 1,979 correct guesses exploited common passwords; 65 fell in both categories. For the common password guesses, 1,705 unique users were affected which was only 0.7% of all users and only 696 were valid at the time the passwords were checked.

We correctly guessed an additional 362 passwords that were active for less than one hour, but neither included them in the numbers above nor in subsequent analyses.

Interestingly, while we only correctly guessed 6 passphrases (containing 18+ characters) based on password reuse, we correctly guessed 17 based on common passwords. Next, we provide a detailed analysis of our results, focusing on reused passwords.

3.1 A Longitudinal Perspective

Our university's time-stamped password history database gave us a unique (compared to prior work) two-decade retrospective look at our university's longitudinal vulnerability to password-guessing attacks. Figure 2 shows, over time, the number of accounts for which a password we correctly guessed was active (i.e., the user's current password), comparing reused passwords and common passwords. The steep yearly increase coincides with incoming students creating accounts, suggesting that we guessed a number of users' first passwords at the university. The number of active passwords that we correctly guessed increased steadily until late 2014.

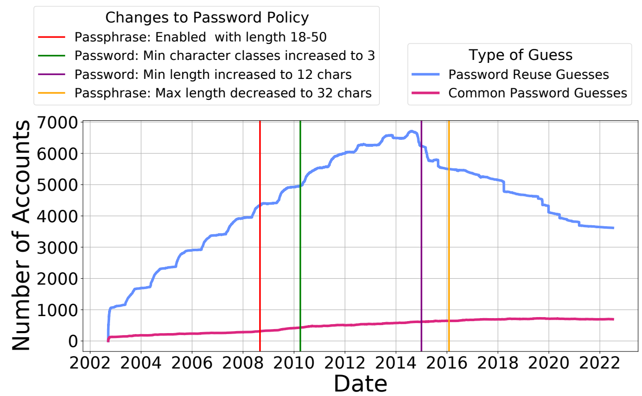

Figure 2: At the time indicated on the x-axis, the number of accounts actively using a password we correctly guessed.

Table 6: Policy compliance of correct guesses.

Policy	Password Reuse Guesses: Passwords	Password Reuse Guesses: Users	Common Password Guesses: Passwords	Common Password Guesses: Users
Password (Current)	1,417	1,104	849	697
Password (Pre-2015)	7,011	5,984	1,365	1,169
Password (Pre-2010)	12,224	10,179	1,962	1,689
Passphrase (Current)	6	6	17	16

At that point, **the minimum password length increasing from eight to twelve characters coinciding with a steep drop in the number of active passwords correctly guessed based on password reuse**. That drop continues through the present. We found over five times as many leaked passwords compliant with the older policy compared to the new policy. Thus, our university's relatively stringent and unique new password-composition policy likely contributed to this drop. While the majority of our top individual service breaches (Table 7) became public around 2016, with Chegg from 2019 and LiveJournal from 2020 (and breach compilations peaking around 2019), the decrease in recent major public breaches may have also played a role in the decline. Whereas 12,224 correct guesses based on password reuse complied with the pre-2010 policy and 7,011 complied with the 2010–2015 policy (requiring three, not two, character classes), only 1,417 complied with the current policy (minimum length of 12 characters). While there was no explicit requirement that affiliates update their password when the new policy went into effect in 2014, a minority of users (including those at our medical center) at the time were subject to periodic password expiration, which may have contributed to the quick drop.

As Table 6 shows, password reuse was a far greater threat than common passwords. Furthermore, we made more correct guesses for older and less restrictive password-composition policies, but only a few for passphrase policies.

Figure 3 shows for how long correctly guessed passwords remained active. **Credentials we correctly guessed were active for a median of 6.2 years**, with a maximum of 19.8 years.

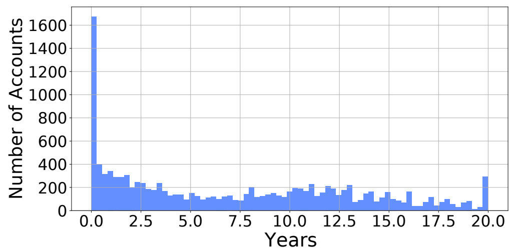

Figure 3: The length of time for which correctly guessed passwords (including those currently valid) had been active.

Table 7: Top **individual service breaches** for guessing.

Name of Service	Reported Date of Breach	Total # Leaked Passwords	Total # Correct Guesses	# Guesses Currently Valid
LinkedIn	May 2012	195,110	2,433	533
Chegg	Apr 2018	108,702	1,938	498
LiveJournal	Jan 2017	58,632	979	215
Dropbox	Jul 2012	41,013	903	287
MySpace	Jul 2008	1,976	767	108
Twitter*	Jun 2016	74,970	396	124
Last.fm	Sep 2012	626	217	17
Neopets	May 2013	57,665	129	45
Gmail*	Jan 2014	4,002	106	38
Zynga	Sep 2019	3,998	106	38
Coupon Mom & Armor Games*	Feb 2014	18,533	99	33
Evony	Jun 2016	34,649	84	34
Zoosk*	Jan 2011	73,527	64	24
Fling	Mar 2011	67,915	62	23
Canva	May 2019	3,971	49	13
Stratfor	Dec 2011	5,149	44	15
Brazzers	Apr 2013	4,457	40	11
Yahoo	Jul 2012	4,251	40	7
Wattpad	Jun 2020	4,655	39	16
Mate1	Feb 2016	40,675	39	10
Forbes	Feb 2014	2,137	28	9
Comcast	Nov 2015	3,073	26	10
VK	Jan 2012	35,072	25	8
Ashley Madison	Jul 2015	17,029	23	12

** Not confirmed by the service provider; the leak may be from phishing.*

Notably, 7,268 correctly guessed credentials were active beyond when they were no longer compliant with the active composition policy. At the time of analysis in 2022, a total of 2,071 correct guesses only met the pre–2015 policy, while 1,525 only met the pre-2010 policy. We correctly guessed multiple passwords for 1,577 users (15.5%). In fact, for one user, we correctly guessed 9 passwords. When we correctly guessed multiple passwords for a single user, they were typically created successively.

3.2 Sources of Leaked Passwords

Ultimately, **71 different individual service breaches and all 12 breach compilations we tested bootstrapped at least one correct guess**. Table 7 summarizes the individual service breaches that bootstrapped the most correct guesses. The full results can be found in our extended version [56]. Notably, the breaches of LinkedIn, Chegg, LiveJournal, Dropbox, and MySpace each bootstrapped over 500 correct guesses, while 34 different breaches bootstrapped at least ten correct guesses.

Table 8: Correct guesses from **breach compilations**.

Breach Compilation	Date Made Public	Total # Leaked Passwords	Total # Correct Guesses	# Guesses Currently Valid
1.4B Breach Compilation	Nov 2017	1,561,449	7,715	2,301
Collection #2	Jan 2019	2,358,605	7,591	2,322
Big Database Combo List	Jan 2019	2,307,980	7,499	2,295
XSS.is 13B Account Leak	Jan 2019	2,112,070	6,960	2,104
Anti Public Combo List	Dec 2016	1,428,024	5,366	1,576
Collection #4	Jan 2019	1,397,357	5,164	1,622
Collection #1	Jan 2019	883,075	3,591	1,153
Exploit.In Combo List	Oct 2016	631,361	2,956	857
Collection #5	Jan 2019	621,260	2,595	843
Collection #3	Jan 2019	466,580	2,468	827
AP MYR & ZABUGOR	Jan 2019	346,423	1,260	383
Onliner Spambot	Aug 2017	1,550	436	82

Analogously, Table 8 reports on breach compilations. Eleven of the twelve compilations bootstrapped at least 1,000 correct guesses, though there was substantial overlap between them.

Figure 4 traces the top individual service breaches and all breach compilations temporally, showing the number of accounts active at a given time whose credentials were correctly guessed from that source. Notably, this graph highlights how this vulnerability compares to when each breach occurred and was made public. Individual service breaches typically reached their vulnerability peak around when the breach occurred, whereas the release of breach compilations trailed their vulnerability peak by a few years. The steep drops in the graph correspond to passwords reset by ITS based on suspicious activity (see Section 3.6).

Even after a breach was made public, many accounts remained vulnerable for years. Most dramatically, at the time LinkedIn was breached, there were 1,415 active accounts that we eventually correctly guessed using leaked passwords from LinkedIn. **It took seven and a half years for even half of those vulnerable passwords to be changed.**

Before the corresponding leaked password appeared in any of our data sources, 5,398 of our correct guesses were no longer active, meaning those accounts may not have ever been vulnerable in practice. That said, attackers may have additional breaches we did not. In contrast, 5,915 correctly guessed passwords were created before appearing publicly, while 934 were created at our university after appearing publicly. Unfortunately, credential checking services like HIBP are typically employed when users create a password, so they would miss the (more common) former case.

A figure that we included in our extended version [56] shows the distribution of the time of vulnerability. The longest was over 14 years, and the mean was just under 5 years when only considering passwords that were active when the corresponding breach became public.

We found 7,006 (57.2%) of our correct guesses only in plaintext, 1,806 (14.7%) only as hashes, and 3,435 (28.0%) as both. The most common hash functions that yielded correct guesses were unsalted MD5 (2,393 correct guesses), unsalted SHA-1 (2,201), and bcrypt (1,025).

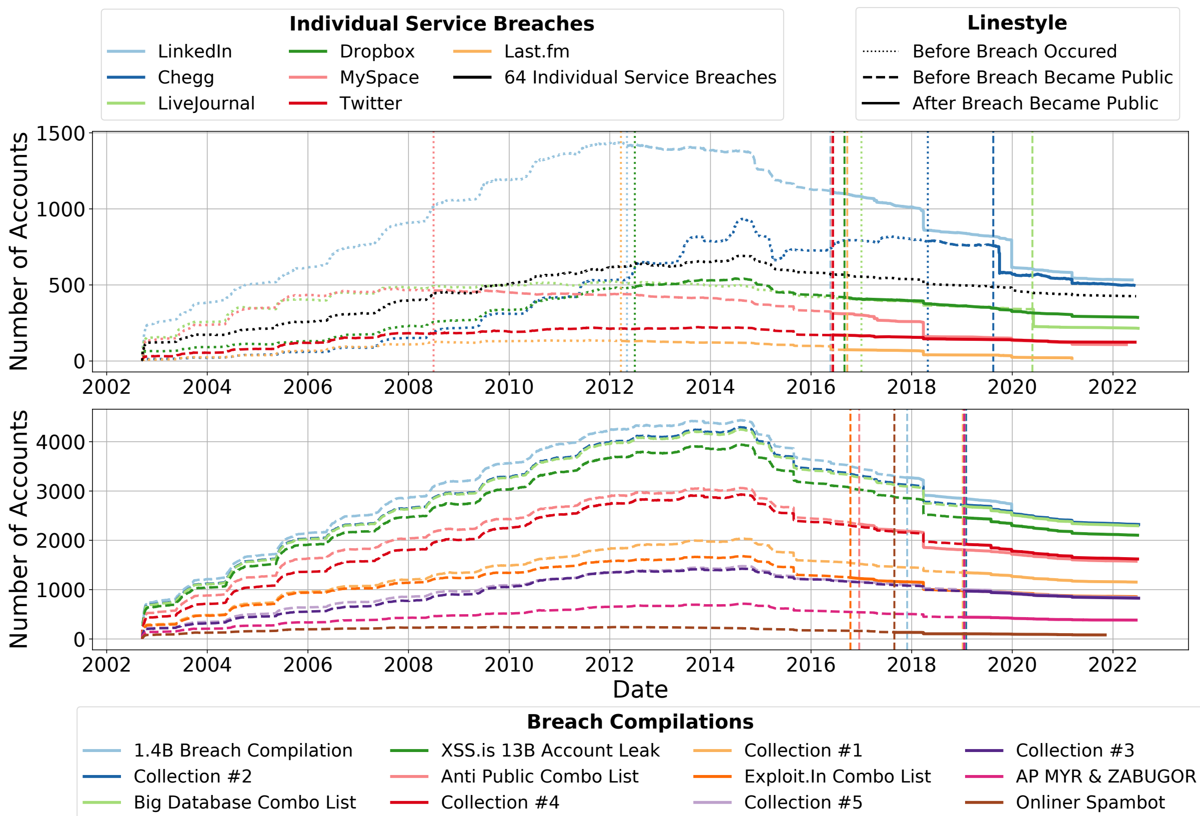

Figure 4: Number of accounts vulnerable over time from individual service breaches (top) and breach compilations (bottom).

Consistent with prior research [49], few survey respondents were aware that their data had been in a data breach or that the stolen passwords were similar to their university credentials.

3.3 Email- and Username-Based Matching

Exact email matches were by far the most successful strategy, accounting for 5,653 correct guesses. Similar email matches resulted in 7,463 correct guesses, and usernames 1,857. The latter two strategies are prone to false positives. Notably, exact email matches accounted for only 18,205 leaked credentials (versus 2,719,214 and 530,391, respectively). Emphasizing the high probability of guesses derived from exact email matches, we correctly guessed a password for 32.0% of users with an exact email match. The same was true for only 4.7% of users with a similar email match and 1.5% of those with a username match. By comparison, as Figure 5 shows, survey respondents most self reported commonly expected that each of these three matching strategies would match at most 25% of their non-university accounts. While exact email matches resulted in the most effective guesses, respondents reported them as least likely to match their other accounts.

Overall, email matches from 1,408 different domain names bootstrapped a correct guess. As shown in Table 9, `uchicago.edu` was by far the most common, followed by `gmail.com` and `yahoo.com`. In the long tail of domains, we observed many `.edu` domains from other institutions, indicating users who reused their password while at multiple academic institutions. We also observed a smaller number of correct guesses for other university-related domains (e.g., the business school's domains), as well as other services from the city in which our institution is located.

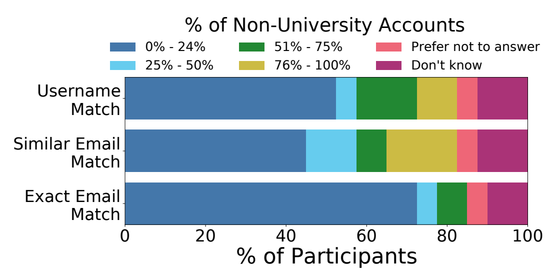

Figure 5: Survey respondents' estimates of the fraction of their accounts that could be matched to their university account.

3.4 Affiliations

The majority of users whose passwords we correctly guessed are currently alumni, as shown in Table 10. This is unsurprising since alumni vastly outnumber current students

Table 9: Most frequent email domains for correct guesses. The three domains with asterisks relate to the business school.

Email Domain	#	Email Domain	#
uchicago.edu	5,020	chicagogsb.edu*	218
gmail.com	3,136	gsb.uchicago.edu*	217
yahoo.com	1,295	chicagobooth.edu*	213
hotmail.com	988	alumni.uchicago.edu	185
mail.ru	383	ya.ru	176
aol.com	292	rambler.ru	105
comcast.net	238	sbcglobal.net	101
yandex.ru	236		

Table 10: Vulnerable users by current affiliation.

Affiliation	Ever Vulnerable (% of Affiliates)		Currently Vulnerable (% of Affiliates)	
Alumni	7,875	(6.2%)	2,607	(2.1%)
None	1,453	(3.5%)	912	(2.2%)
Employees	349	(3.4%)	13	(0.1%)
Students	295	(1.2%)	66	(0.3%)
Faculty	92	(5.0%)	4	(0.2%)
Other Academic	69	(2.8%)	3	(0.1%)
Other	53	(4.2%)	13	(1.0%)

and staff. That said, alumni also had the highest percentage of users (6.2%) that had at least one correctly guessed password, which is consistent with prior work [58]. Comparatively, current students had the lowest percentage (1.2%). Notably, alumni and faculty have likely held their accounts longer than current students, giving them more time to reuse credentials.

Individual service breaches do not necessarily impact particular types of affiliates equally. Most clearly, Table 11 shows the vulnerability of different types of affiliates to the LinkedIn (2012) and Chegg (2018) data breaches. Among all students for whom we correctly guessed a password, 41.4% had a correct guess derived from a password in the Chegg breach, versus only 2.2% of faculty. Conversely, among all faculty for whom we correctly guessed a password, 54.3% had a correct guess derived from a password in the LinkedIn breach, versus only 11.2% of students. Given that Chegg is a homework-focused site and LinkedIn is a professional social network, these differences make intuitive sense.

3.5 Credential Tweaking Algorithms

Most commonly, our correct guess was simply the leaked password verbatim (i.e., without tweaking). In our case, 6,694 correct guesses (54.7%) exactly matched the leaked password, while the remaining 5,553 (45.3%) required tweaking. **The most successful tweaks were toggling the first character's case (≈11% of correct guesses) and appending either '!' (≈4%) or '1' (≈2%).** These are all common coping strategies for complying with policies that demand uppercase characters, symbols, and digits [20,70], lending credence to NIST SP 800-63B dropping such requirements [23].

Table 11: Vulnerability to Chegg and LinkedIn breaches by current affiliation, including the percentage of vulnerable affiliates of that type who were vulnerable due to that breach.

Affiliation	Chegg (% of Vulnerable)		LinkedIn (% of Vulnerable)	
Alumni	1,264	(16.1%)	1,494	(19.0%)
None	147	(10.1%)	339	(23.3%)
Employees	36	(10.3%)	123	(35.2%)
Student	122	(41.4%)	33	(11.2%)
Faculty	2	(2.2%)	50	(54.3%)
Other Academic	4	(5.8%)	22	(31.9%)
Other	13	(24.5%)	7	(13.2%)

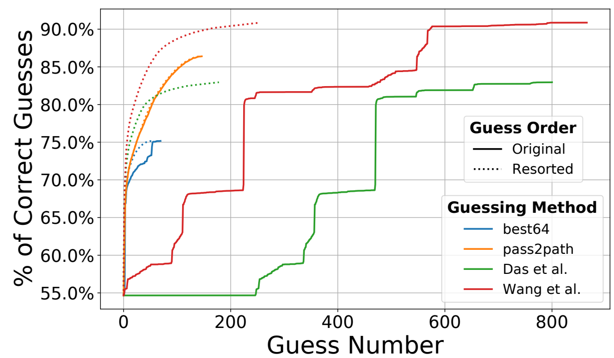

Figure 6: Comparison of the credential tweaking approaches.

Figure 6 compares the four credential tweaking approaches tested (Section 2.4). The y-axis starts at 54.7% because a reasonable attacker would first guess the leaked password verbatim. It ends near 90% because none of the four approaches individually captured all correct guesses made by the union.

As configured "out of the box," the best source of guesses was the *pass2path* approach from Pal et al. [58], which captured 86.4% of correct guesses.While pass2path is computationally very expensive and requires training data and policy adjustments, the comparatively easy and straightforward `best64.rule` approach captured 75.2% of correct guesses.

The two heuristics-based approaches performed well in terms of coverage but less well in terms of the effectiveness of initial guesses. The Das et al. [10] and Wang et al. [80] approaches respectively captured 83% of correct guesses.These approaches are highly similar algorithmically, though Wang et al. more frequently applies two transformations at once (often at the beginning and the end of the string), leading to more correct guesses, as well as more guesses in total. In practice, rate-limiting [21,46] and risk-based authentication [88] limit guessing. For instance, NIST recommends limiting the number of failed attempts on a single account to 100 within any 30-day period [23]. If we apply these recommendations, the best performing algorithms are *pass2path*, `best64.rule`, and Wang et al., with 84.6%, 75.2%, and 61.9% coverage, initially seeming to confirm past work [58].

However, the order of rules in the Das et al. and Wang et al. papers seems not to have been optimized. Applying a perfect

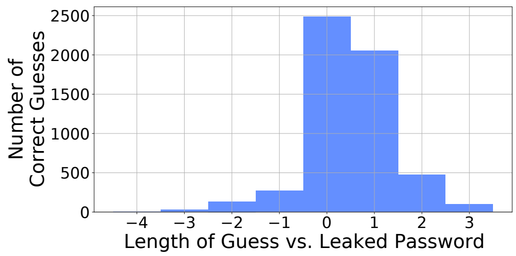

Figure 7: Difference in length between the leaked password and the correct guess, excluding verbatim reuse. Positive numbers indicate a guess longer than the leaked password.

knowledge attacker model [6] that always guesses in the most effective order, the Wang et al. approach and, at least for a smaller number of guesses, the Das et al. approach appear more effective than pass2path as shown by the dotted lines in Figure 6. Notably, the reordered Wang et al. and Das et al. approaches are lower bounds on their effectiveness. Whereas *pass2path*'s guesses are password-specific, Wang et al. and Das et al. simply specify the transformation. To minimize the possibility of re-identification, our metadata does not capture which preceding transformations do not modify the leaked password or comply with a policy.

As shown in Figure 7, correct guesses (post-tweaking) were more often longer than the leaked password, as opposed to shorter. That said, the most common difference in length between the leaked password and the correct guess was 0 (i.e., a modification that does not change the length). This held true for the former password-composition policies. For the current policy, though, almost twice as many correct guesses were one character longer than the leaked password.

3.6 Exploited Passwords

When they notice suspicious activity on an account indicating an apparent compromise, our ITS team locks the account, forces a password reset, and records these actions in a time-stamped log. Unlike in prior work, we were thus able to compare our correct guesses with possible exploitation by attackers. **Apparent compromises were most likely for exact email matches and verbatim reuse.**

Among correct guesses where the user's password change was mandated by ITS due to an apparent compromise, 83.6% were found verbatim in a leak (i.e., without tweaking); this was only true for 47.0% of password resets initiated by the user. Looking at the numbers a different way, 42.4% of our correct guesses based on verbatim reuse were associated with an apparent compromise, while only 11.3% of our tweaked correct guesses were. We observed a similar trend for exact email matches. Among correct guesses where the user's password change was mandated by ITS (i.e., apparent compromises), 79.2% were from exact email matches. While we had hypothesized that leaked passwords appearing in plaintext (vs. hashed) would follow a similar pattern, the effect was more muted. In total, 30.2% of correct guesses where we found the leaked password in plaintext, 24.6% of correct guesses where we only found a hash, were associated with apparent compromise. In other words, cracking hashes did not seem to be as much of a barrier for attackers as credential tweaking or inexact account matching. In sum, among apparent compromises, 60.7% were an exact email match whose password was found verbatim in plaintext.

Table 12: Days when 25+ accounts whose passwords we guessed exhibited suspicious activity and associated breaches.

Date	#	Associated Breaches and Compilations (#)
03/26/18	291	1.4B Breach (291), Anti Public (289), Big Database (289), Collection #2 (289), XSS.is 13B (281), Collection #4 (153)
12/27/19	206	1.4B Breach (206), **LinkedIn (180)**
09/30/19	134	**Chegg (134)**
08/28/15	125	Big Database (117), Collection #2 (117), XSS.is 13B (117), Anti Public (110), 1.4B Breach (107), Exploit.In (95), Collection #1 (93), Collection #4 (90)
06/02/20	115	**LiveJournal (115)**
03/09/21	113	1.4B Breach (59)
08/27/15	61	Big Database (57), Collection #2 (57), Anti Public (55), XSS.is 13B (54), 1.4B Breach (47), Collection #1 (39), Collection #4 (39), Exploit.In (36)
07/30/19	61	Collection #2 (58), Big Database (56), XSS.is 13B (52), Collection #4 (50)
04/04/17	36	Anti Public (36), Big Database (36), Collection #2 (36), 1.4B Breach (35), XSS.is 13B (34), Collection #4 (21), Exploit.In (20)
09/25/19	26	**Chegg (26)**
05/23/16	25	1.4B Breach (25), Big Database (23), Collection #2 (23), XSS.is 13B (22), Anti Public (19), Collection #4 (18), **Last.fm (16)**
09/16/20	25	Big Database (18), Collection #2 (18), XSS.is 13B (18), 1.4B Breach (17), Anti Public (16), Collection #4 (13)

On 29 separate days over the last eight years, ITS observed suspicious activity (forcing password resets) for at least ten accounts whose passwords we guessed. Table 12 shows the 12 days with the most resets. Five of these days are highly associated with specific individual service breaches: LinkedIn, Chegg, LiveJournal, Chegg again, and Last.fm. Some of this exploitation was quick. For instance, all apparently compromised accounts on September 30th, 2019, were found in the Chegg breach not long after it was added to HIBP on August 16th, 2019. In the survey, several respondents mentioned that they did not remember even creating or having a Chegg account, making this apparent exploitation all the more dangerous. Similarly, all apparently compromised accounts on June 2nd, 2020, were found in the LiveJournal breach, which was added to HIBP on May 26th, 2020. On some other dates, all passwords were found in the 1.4B Breach Compilation.

3.7 User Understanding and Attitudes

Our survey provided additional insight into affiliates' perceptions. While none of the 40 respondents recalled any unauthorized access to their university account, 23 (57.5%) knew that a non-university account had been compromised in a data breach and nine (22.5%) believed someone had actually gained access to a non-university account. Respondents with a current university affiliation were both more concerned with the possibility of someone gaining access to their account and likely to consider their university account important.

Only two of the 28 respondents whose password was guessed from one or more breach compilations even reporting having heard of such compilations. Of respondents asked about individual data breaches, eight (42.1%) did not even know they had an account for that service. Notably, seven of those eight were from Chegg. Five participants that knew they had the account knew the passwords were similar, and six knew their credentials had been included in a data breach.

Of the 27 respondents forced to reset their password, 12 (44%) said the password we correctly guessed was exactly the same as a password they still used on yet another unrelated account. Even after being forced to reset their password, nine (33%) of these respondents nonetheless reported resorting to verbatim password reuse for their new password.

The survey also asked about respondents' comfort with compromised credential checking. Participants were most comfortable with ITS checking if their credentials appeared in breaches either collected themselves or via credential-checking services; see our extended version [56] for more details. Respondents were less comfortable with ITS or academics trying to guess their password, though most respondents were comfortable with all of these scenarios.

4 Related Work

In this section, we briefly highlight key prior work.

Password Reuse. Numerous studies [3, 14, 41, 64, 72, 84] have reported that users reuse passwords. The account value, frequency of use, composition policy, account matching, guessing methods, and data sources all vary across prior work, resulting in different estimated rates of password reuse.

Password Tweaking. While many users reuse passwords verbatim across accounts, some make modifications. Das et al. [10] developed an algorithm that could guess 30% of non-identical password pairs within 100 attempts from a set of 6,077 unique users. Later, Wang et al. [80] developed an algorithm based on a dataset of 107 online services with 7,196,242 pairs of leaked passwords. They guessed 46.5% of the modified passwords within 100 guesses. In 2019, Pal et al. [58] developed the `pass2path` machine learning model, which guessed 15.8% of modified passwords in 1,000 guesses.

Users' Knowledge of Data Breaches. User studies have found that users often do not know their information has appeared in a data breach, even if they had heard of the data breach occurring [49, 91]. Generally, users have a good understanding of what data breaches are, but often lack a concrete understanding of why they are affected [28, 37]. While users want to be notified immediately of data breaches [37], current notifications do not cause users to report taking adequate actions and can lead to misconceptions [22, 31, 90, 92].

Compromised Credential Checking. Due to the risks posed by password reuse, in 2017 NIST updated their digital identity guidelines to require that new passwords be checked against "passwords from breach corpuses" [23]. Hunt developed the HIBP "Pwned Passwords" API [33], enabling organizations to check whether passwords appear in hundreds of data breaches. This API is used by many websites and products [13], including our own university (starting in late 2019). Outside of HIBP, companies like Google [73], Mozilla [55], and Apple [12] have developed their own compromised credential checking (C3) APIs. However, C3 services must prevent attackers from extracting breached credentials. Recent work [40, 43, 44, 59, 82, 83] aims to improve these protocols.

Supporting Users. Users are confronted with demanding password composition policies and requirements [41, 50, 51, 66, 81, 86, 89]. Users adopt various coping strategies, including using easy-to-memorize (and thus easy-to-guess) passwords or reusing passwords [16, 18, 26, 50, 61, 68, 69, 74, 75, 85]. Our work confirms the prevalence of these strategies. Password managers have long been recommended for maintaining a unique password on each account. However, adoption remains low [62] and features like random password generation often go unused [1, 30, 47, 48]. Enabling 2FA adds a layer of security even if the password is compromised. However, 2FA has its own problems [8], and voluntary adoption is also low. Companies now offer services that reduce friction in changing passwords [54, 57, 63] or hide a user's real email address, making it harder for attackers to match accounts [2, 39].

5 Discussion and Conclusions

We presented a 20-year analysis of our university's vulnerability to credential-guessing attacks. Our approach using a large number of individual service provider breaches and breach compilations let us understand how specific service provider breaches impact vulnerability over time and how the different sources connect to actual exploitation of accounts.

Contextualizing our results, we find slightly lower rates of reuse than previous studies, but major differences in methodology and password composition policies make comparisons difficult. We provide a comparison table in our extended version [56]. Prior work on Cornell University accounts by Pal et al. [58] found between 2.6% and 8.4% of passwords were vulnerable to guessing attacks based on password reuse. Sanusi et al. [65] found a lower rate of reuse when using `pass2path` at two universities. Studying a different sample, Thomas et al. found that 7.5% of Google users had a password in their set of data breaches [72]. In our study, we found 5.0% of current users were vulnerable based on exact email matching, and 2.1% on similar email matching. This lower rate might in part be related to differences in password policies (8 vs. 12 character minimum). Our work adds to this limited literature by uniquely *longitudinally* analyzing the impact of a far more comprehensive array of data sources, matching strategies, tweaking algorithms, hash cracking, and correlations with apparent account compromises.

Perspective from the University's IT Security Team: In discussing the results with our contacts at ITS, they expressed surprise at the raw number of passwords that we were able to guess and how well the basic transformations worked. Conversely, they expected the vast majority of our correct guesses to be for very old accounts, and they were surprised that we were also able to guess more recent accounts. While ITS cares about the security of alumni accounts, they are less of a priority than, for instance, current faculty accounts.

From their side, the collaboration took approximately 100 hours of work. While actually checking if the credentials were correct took 20-25 hours, locking accounts and gathering other information that was returned to the academic researchers took much longer. Running into corner cases that has built up over the years and dealing with the scale of the data were also hurdles that ITS had to overcome.

Our ITS team's hope is to move the university away from passwords entirely in the coming years, so repeating this sort of analysis would provide limited value. For organizations that are further away from potential transitions to passwordless authentication or that do not have 2FA set up, our contacts felt the proactive checking we performed in this study could be more advantageous. This type of checking might also be useful for identifying accounts to monitor more closely.

Based on our findings, we recommend that defenders:

- **R1** Check for high-risk (i.e., organization-related) breaches
- **R2** Not ignore the long tail of individual service breaches
- **R3** Check for *similar* email matches and username matches, not only exact email matches
- **R4** Save computational resources by starting with heuristic tweaking algorithms, not ones based on machine learning
- **R5** Crack hashes to protect against motivated attackers
- **R6** Implement processes to expire unused accounts

We next detail how our results motivated these specific (numbered) recommendations.

Vulnerable passwords come from an array of individual service breaches and breach compilations [R1]. High-profile leaks like LinkedIn enabled a significant number of correct guesses. Further, we observed a high correlation with leaks from academic-related services like Chegg that are of particular interest to attackers trying to compromise academic accounts [85]. There was a very quick turnaround between the Chegg data breach becoming public and direct reuse of Chegg passwords being exploited at our university. Temporary additional defenses for users with exact email matches in the breach may help stave off such rapid attacks.

Smaller data breaches can pose significant risks to accounts [R2]. While large individual service breaches bootstrapped our most successful guesses, skipping over smaller individual service data breaches or large (poorly formatted) compilations may cause defenders to miss at-risk accounts. Unfortunately, processing breaches requires defenders' time.

Adequately protecting user accounts will require accounting for looser matching, transformations, and cracking hashes [R3, R4, R5]. While exact email matches accounted for one portion of vulnerable accounts (4,585 users), another meaningful portion were similar email matches from non-university domains (6,951 users). This implies that checking for password reuse with only exact email matches may not be enough to protect users from motivated attackers. Furthermore, users reuse passwords verbatim more often than they marginally tweak passwords: 55% of correct guesses exactly matched the original password. The remaining 45% of correct guesses required transformations, with the most successful being the classic strategies to comply with composition policies: capitalizing the first character or appending '!' or '1' [75]. Light-weight, heuristic-based transformation, if more carefully ordered, seems comparable to computationally heavy deep-learning-based approaches, though all credential-tweaking approaches uniquely guessed some passwords. In the same vein, 14.7% of our successful guesses were found only as hashes, with unsalted MD5, unsalted SHA-1, and bcrypt accounting for most of those guesses, and similar email matches accounted for the largest number of correctly guessed passwords (but were also much more prone to false positives).

Passwords are at risk for long periods of time; users may not know about the risk to their account [R6.] Passwords we correctly guessed were active for a median of 6 years. Further, the number of accounts that appear to be reusing passwords increased annually up to the end of 2014. Only after our university changed its password policy to increase the minimum length from 8 to 12 characters was there a steep drop in the number of accounts that we identified as reusing passwords. This further confirms the finding that users often do not know that their information has appeared in a data breach [91]. Even when users are informed, they often do not take sufficient action to secure their accounts [49]. Additionally, we found that users may not even be aware that they had accounts on breached sites to begin with. Many accounts remained vulnerable for years, including as student accounts transitioned to alumni accounts. Some were actually exploited years after the breach. Many organizations currently do not expire passwords [19], but perhaps expiration over long periods should be considered. More work into securing legacy accounts is necessary from the research community.

Requiring longer passwords can have temporary protective effects against password reuse attacks. With the decision of our institution's IT department to increase the minimum length of newly created passwords, we observed a steady decline in the number of vulnerable accounts over the past 7 years. We analyzed many leaked passwords that were short, indicating that when account value is high, enforcing longer passwords can provide more protection. However, longer passwords will only provide temporary protections at the cost of burdening users.

Acknowledgments

We are grateful to the UChicago administration and IT Services staff for their support. We thank the NSF-supported *Chameleon* testbed [38], as well as *CynoSure Prime* and friends—especially Sein Coray (*s3in!c*), Sam Croley (*Chick3nman*), Paul D. Ouderkirk (*pdo*), Robert Reif (*winxp5421*), Michael Sprecher (*hops*), and Royce Williams (*TychoTithonus*)—for help cracking hashes. We also thank Lydia Filipe for survey feedback. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2140739.

References

[1] Apple, Inc. Password Manager Resources, May 2020. https://opensource.apple.com/projects/password-manager-resources/

[2] Apple, Inc. What is Hide My Email?, September 2021. https://support.apple.com/en-us/HT210425

[3] Daniel V. Bailey, Markus Dürmuth, and Christof Paar. Statistics on Password Re-use and Adaptive Strength for Financial Accounts. In *Proc. SCN*, 2014.

[4] Jeremiah Blocki, Ben Harsha, and Samson Zhou. On the Economics of Offline Password Cracking. In *Proc. IEEE S&P*, 2018.

[5] Jeremiah Blocki and Wuwei Zhang. DALock: Password Distribution-Aware Throttling. In *Proc. PETS*, 2022.

[6] Joseph Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *Proc. IEEE S&P*, 2012.

[7] Joseph Bonneau, Cormac Herley, Paul C. Van Oorschot, and Frank Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *Proc. IEEE S&P*, 2012.

[8] Jessica Colnago, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Lorrie Faith Cranor, and Nicolas Christin. "It's Not Actually That Horrible": Exploring Adoption of Two-Factor Authentication at a University. In *Proc. CHI*, 2018.

[9] Sam Croley ("Chick3nman"). Abusing Password Reuse at Scale: Bcrypt and Beyond, August 2018. https://www.youtube.com/watch?v=5su3_Py8iMQ

[10] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The Tangled Web of Password Reuse. In *Proc. NDSS*, 2014.

[11] Serge Egelman, Joseph Bonneau, Sonia Chiasson, David Dittrich, and Stuart Schechter. It's Not Stealing If You Need It: A Panel on the Ethics of Performing Research Using Public Data of Illicit Origin. In *Proc. WECSR*, 2012.

[12] Filipe Espósito. iCloud Keychain Now Alerts Users about Leaked Passwords, July 2020. https://9to5mac.com/2020/07/04/ios-14-icloud-keychain-now-alerts-users-about-leaked-passwords-more/

[13] Michael Fey. Watchtower Notifications: Timely Security Alerts for the Websites You Use, July 2020. https://blog.1password.com/announcing-watchtower-notifications/

[14] Dinei Florêncio and Cormac Herley. A Large-scale Study of Web Password Habits. In *Proc. WWW*, 2007.

[15] Dinei Florêncio, Cormac Herley, and Paul C. Van Oorschot. An Administrator's Guide to Internet Password Research. In *Proc. LISA*, 2014.

[16] Dinei Florêncio, Cormac Herley, and Paul C. Van Oorschot. Password Portfolios and the Finite-Effort User: Sustainably Managing Large Numbers of Accounts. In *Proc. USENIX Security*, 2014.

[17] Dinei Florêncio, Cormac Herley, and Paul C. Van Oorschot. Pushing on String: The "Don't Care" Region of Password Strength. *CACM*, 59(11):66–74, October 2016.

[18] Shirley Gaw and Edward W. Felten. Password Management Strategies for Online Accounts. In *Proc. SOUPS*, 2006.

[19] Eva Gerlitz, Maximilian Häring, and Matthew Smith. Please do not use !?_ or your License Plate Number: Analyzing Password Policies in German Companies. In *Proc. SOUPS*, 2021.

[20] Maximilian Golla and Markus Dürmuth. On the Accuracy of Password Strength Meters. In *Proc. CCS*, 2018.

[21] Maximilian Golla, Theodor Schnitzler, and Markus Dürmuth. "Will Any Password Do?" Exploring Rate-Limiting on the Web. In *Proc. WAY*, 2018.

[22] Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa Redmiles, and Blase Ur. "What was that site doing with my Facebook password?" Designing Password-Reuse Notifications. In *Proc. CCS*, 2018.

[23] Paul A. Grassi, James L. Fenton, and William E. Burr. Digital Identity Guidelines – Authentication and Lifecycle Management: NIST Special Publication 800-63B, June 2017.

[24] Andy Greenberg. Hackers are passing around a megaleak of 2.2 billion records, January 2019. https://www.wired.com/story/collection-leak-usernames-passwords-billions/

[25] Hana Habib, Jessica Colnago, William Melicher, Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Password Creation in the Presence of Blacklists. In *Proc. USEC*, 2017.

[26] S.M. Taiabul Haque, Matthew Wright, and Shannon Scielzo. A Study of User Password Strategy for Multiple Accounts. In *Proc. CODASPY*, 2013.

[27] Benjamin Harsha, Robert Mortona, Jeremiah Blocki, John Springer, and Melissa Dark. Bicycle Attacks Considered Harmful: Quantifying the Damage of Widespread Password Length Leakage. *Computers & Security*, 100(1):233–249, January 2021.

[28] Zahra Hassanzadeh, Robert Biddle, and Sky Marsen. User Perception of Data Breaches. *IEEE Trans. Prof. Commun.*, 64(4):374–389, October 2021.

[29] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, and Fernando Perez-Cruz. PassGAN: A Deep Learning Approach for Password Guessing. In *Proc. ACNS*, 2019.

[30] Nicolas Huaman, Sabrina Amft, Marten Oltrogge, Yasemin Acar, and Sascha Fahl. They Would Do Better If They Worked Together: The Case of Interaction Problems between Password Managers and Websites. In *Proc. IEEE S&P*, 2021.

[31] Yue Huang, Borke Obada-Obieh, and Konstantin Beznosov. Users' Perceptions of Chrome Compromised Credential Notification. In *Proc. SOUPS*, 2022.

[32] Troy Hunt. *Have I Been Pwned?* – Check If Your Email Has Been Compromised in a Data Breach, December 2013. https://haveibeenpwned.com

[33] Troy Hunt. *Have I Been Pwned?* – Pwned Passwords v3 is Now Live!, July 2018. https://www.troyhunt.com/pwned-passwords-v3-is-now-live/

[34] Troy Hunt. *Have I Been Pwned?* – Pwned Websites, August 2022. https://haveibeenpwned.com/PwnedWebsites

[35] Marcello Ienca and Effy Vayena. Ethical Requirements for Responsible Research with Hacked Data. *Nature Machine Intelligence*, 3(9):744–748, September 2021.

[36] Saul Johnson, João F. Ferreira, Alexandra Mendes, and Julien Cordry. Skeptic: Automatic, Justified and Privacy-Preserving Password Composition Policy Selection. In *Proc. AsiaCCS*, 2020.

[37] Sowmya Karunakaran, Kurt Thomas, Elie Bursztein, and Oxana Comanescu. Data Breaches: User Comprehension, Expectations, and Concerns with Handling Exposed Data. In *Proc. SOUPS*, 2018.

[38] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. Lessons Learned from the Chameleon Testbed. In *Proc. ATC*, 2020.

[39] M.J. Kelly. Firefox Relay Protects Your Email Address from Hackers and Spammers, June 2020. https://blog.mozilla.org/en/products/firefox/firefox-relay/

[40] Dmitry Kogan and Henry Corrigan-Gibbs. Private Blocklist Lookups with Checklist. In *Proc. USENIX Security*, 2021.

[41] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. Of Passwords and People: Measuring the Effect of Password-Composition Policies. In *Proc. CHI*, 2011.

[42] Kevin Lee, Sten Sjöberg, and Arvind Narayanan. Password Policies of Most Top Websites Fail to Follow Best Practices. In *Proc. SOUPS*, 2022.

[43] Jie Li, Yamin Liu, and Shuang Wu. Pipa: Privacy-Preserving Password Checkup via Homomorphic Encryption. In *Proc. AsiaCCS*, 2021.

[44] Lucy Li, Bijeeta Pal, Junade Ali, Nick Sullivan, Rahul Chatterjee, and Thomas Ristenpart. Protocols for Checking Compromised Credentials. In *Proc. CCS*, 2019.

[45] Enze Liu, Amanda Nakanishi, Maximilian Golla, David Cash, and Blase Ur. Reasoning Analytically About Password-Cracking Software. In *Proc. IEEE S&P*, 2019.

[46] Bo Lu, Xiaokuan Zhang, Ziman Ling, Yinqian Zhang, and Zhiqiang Lin. A Measurement Study of Authentication Rate-Limiting Mechanisms of Modern Websites. In *Proc. ACSAC*, 2018.

[47] Sanam Ghorbani Lyastani, Michael Schilling, Sascha Fahl, Michael Backes, and Sven Bugiel. "Better managed than memorized?" Studying the Impact of Managers on Password Strength and Reuse. In *Proc. USENIX Security*, 2018.

[48] Peter Mayer, Collins W. Munyendo, Michelle L. Mazurek, and Adam J. Aviv. Why Users (Don't) Use Password Managers at a Large Educational Institution. In *Proc. USENIX Security*, 2022.

[49] Peter Mayer, Yixin Zou, Florian Schaub, and Adam J. Aviv. "Now I'm a bit angry:" Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them. In *Proc. USENIX Security*, 2021.

[50] Michelle L. Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. Measuring Password Guessability for an Entire University. In *Proc. CCS*, 2013.

[51] William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In *Proc. USENIX Security*, 2016.

[52] Daniel Miessler and Community. SecLists: 10-million-password-list, March 2018. https://github.com/danielmiessler/SecLists

[53] Lorenzo Neil, Elijah Bouma-Sims, Evan Lafontaine, Yasemin Acar, and Bradley Reaves. Investigating Web Service Account Remediation Advice. In *Proc. SOUPS*, 2021.

[54] Patrick Nepper. Fix Your Passwords in Chrome With a Single Tap, May 2021. https://blog.google/products/chrome/automated-password-changes/

[55] Nick Nguyen. Introducing Firefox Monitor: Helping People Take Control After a Data Breach, September 2018. https://blog.mozilla.org/en/products/firefox/introducing-firefox-monitor-helping-people-take-control-after-a-data-breach/

[56] Alexandra Nisenoff, Maximilian Golla, Miranda Wei, Juliette Hainline, Hayley Szymanek, Annika Braun, Annika Hildebrandt, Blair Christensen, David Langenberg, and Blase Ur. A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords (Extended Version), 2023. https://www.blaseur.com/papers/uchicagoreuse-extended.pdf

[57] Theresa O'Connor and Ricky Mondello. A Well-Known URL for Changing Passwords, January 2021. https://w3c.github.io/webappsec-change-password-url/

[58] Bijeeta Pal, Tal Daniel, Rahul Chatterjee, and Thomas Ristenpart. Beyond Credential Stuffing: Password Similarity Models using Neural Networks. In *Proc. IEEE S&P*, 2019.

[59] Bijeeta Pal, Mazharul Islam, Marina Sanusi Bohuk, Nick Sullivan, Luke Valenta, Tara Whalen, Christopher Wood, Thomas Ristenpart, and Rahul Chatterjee. Might I Get Pwned: A Second Generation Compromised Credential Checking Service. In *Proc. USENIX Security*, 2022.

[60] Dario Pasquini, Ankit Gangwal, Giuseppe Ateniese, Massimo Bernaschi, and Mauro Conti. Improving Password Guessing via Representation Learning. In *Proc. IEEE S&P*, 2021.

[61] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, and Alain Forget. Let's Go in for a Closer Look: Observing Passwords in Their Natural Habitat. In *Proc. CCS*, 2017.

[62] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Why People (Don't) Use Password Managers Effectively. In *Proc. SOUPS*, 2019.

[63] Jay Peters. Dashlane Is Giving Its One-Click Password Changer a Big Upgrade, March 2021. https://www.theverge.com/2021/3/11/22320467

[64] Sena Sahin and Frank Li. Don't Forget the Stuffing! Revisiting the Security Impact of Typo-Tolerant Password Authentication. In *Proc. CCS*, 2021.

[65] Marina Sanusi, Mazharul Islam, Syed Suleman Ahmad, Michael Swift, Thomas Ristenpart, and Rahul Chatterjee. Gossamer: Securely Measuring Password-based Logins. In *Proc. USENIX Security*, 2022.

[66] Richard Shay, Saranga Komanduri, Patrick Gage Kelley, Pedro Giovanni Leon, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Encountering Stronger Password Requirements: User Attitudes and Behaviors. In *Proc. SOUPS*, 2010.

[67] Jens Steube ("atom") and Community. Official Best64 Challenge Thread, March 2012. https://hashcat.net/forum/thread-1002-post-5284.html#pid5284

[68] Elizabeth Stobert and Robert Biddle. The Password Life Cycle: User Behaviour in Managing Passwords. In *Proc. SOUPS*, 2014.

[69] Leona Tam, Myron Glassman, and Mark Vandenwauver. The Psychology of Password Management: A Tradeoff between Security and Convenience. *Behaviour & Information Technology*, 29(3):233–244, April 2010.

[70] Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Practical Recommendations for Stronger, More Usable Passwords Combining Minimum-Strength, Minimum-Length, and Blocklist Requirements. In *Proc. CCS*, 2020.

[71] Daniel R. Thomas, Sergio Pastrana, Alice Hutchings, Richard Clayton, and Alastair R. Beresford. Ethical Issues in Research Using Datasets of Illicit Origin. In *Proc. IMC*, 2017.

[72] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials. In *Proc. CCS*, 2017.

[73] Kurt Thomas, Jennifer Pullman, Kevin Yeo, Ananth Raghunathan, Patrick Gage Kelley, Luca Invernizzi, Borbala Benko, Tadek Pietraszek, Sarvar Patel, Dan Boneh, and Elie Bursztein. Protecting Accounts From Credential Stuffing With Password Breach Alerting. In *Proc. USENIX Security*, 2019.

[74] Blase Ur, Felicia Alfieri, Maung Aung, Lujo Bauer, Nicolas Christin, Jessica Colnago, Lorrie Faith Cranor, Henry Dixon, Pardis Emami Naeini, Hana Habib, Noah Johnson, and William Melicher. Design and Evaluation of a Data-Driven Password Meter. In *Proc. CHI*, 2017.

[75] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M. Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. "I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab. In *Proc. SOUPS*, 2015.

[76] Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, Darya Kurilova, Michelle L. Mazurek, William Melicher, and Richard Shay. Measuring Real-World Accuracies and Biases in Modeling Password Guessability. In *Proc. USENIX Security*, 2015.

[77] Mathieu Valois, Patrick Lacharme, and Jean-Marie Le Bars. Performance of Password Guessing Enumerators under Cracking Conditions. In *Proc. IFIP SEC*, 2019.

[78] Rafael Veras, Christopher Collins, and Julie Thorpe. A Large-Scale Analysis of the Semantic Password Model and Linguistic Patterns in Passwords. *ACM TOPS*, 24(3):2471–2566, April 2021.

[79] Kathryn Walsh, Faiza Tazi, Philipp Markert, and Sanchari Das. My Account Is Compromised – What Do I Do? Towards an Intercultural Analysis of Account Remediation for Websites. In *Proc. WIPS*, 2021.

[80] Chun Wang, Steve T.K. Jan, Hang Hu, Douglas Bossart, and Gang Wang. The Next Domino to Fall: Empirical Analysis of User Passwords across Online Services. In *Proc. CODASPY*, 2018.

[81] Ding Wang and Ping Wang. The Emperor's New Password Creation Policies. In *Proc. ESORICS*, 2015.

[82] Ke Coby Wang and Michael K. Reiter. How to End Password Reuse on the Web. In *Proc. NDSS*, 2019.

[83] Ke Coby Wang and Michael K. Reiter. Detecting Stuffing of a User's Credentials at Her Own Accounts. In *Proc. USENIX Security*, 2020.

[84] Rick Wash, Emilee Radar, Ruthie Berman, and Zac Wellmer. Understanding Password Choices: How Frequently Entered Passwords are Re-used Across Websites. In *Proc. SOUPS*, 2016.

[85] Miranda Wei, Maximilian Golla, and Blase Ur. The Password Doesn't Fall Far: How Service Influences Password Choice. In *Proc. WAY*, 2018.

[86] Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords. In *Proc. CCS*, 2010.

[87] Daniel Lowe Wheeler. zxcvbn: Low-Budget Password Strength Estimation. In *Proc. USENIX Security*, 2016.

[88] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. What's in Score for Website Users: A Data-Driven Long-Term Study on Risk-Based Authentication Characteristics. In *Proc. FC*, 2021.

[89] Yinqian Zhang, Fabian Monrose, and Michael K. Reiter. The Security of Modern Password Expiration: An Algorithmic Framework and Empirical Analysis. In *Proc. CCS*, 2010.

[90] Yixin Zou, Shawn Danino, Kaiwen Sun, and Florian Schaub. You 'Might' Be Affected: An Empirical Analysis of Readability and Usability Issues in Data Breach Notifications. In *Proc. CHI*, 2019.

[91] Yixin Zou, Abraham H Mhaidli, Austin McCall, and Florian Schaub. "I've Got Nothing to Lose": Consumers' Risk Perceptions and Protective Actions after the Equifax Data Breach. In *Proc. SOUPS*, 2018.

[92] Yixin Zou and Florian Schaub. Beyond Mandatory: Making Data Breach Notifications Useful for Consumers. *IEEE S&P Magazine*, 17(2):67–72, 2019.