Measuring the Risk Password Reuse Poses for a University

Alexandra Nisenoff (*University of Chicago / Carnegie Mellon University*)
Maximilian Golla (*University of Chicago / Max Planck Institute for Security and Privacy*)
Blase Ur (*University of Chicago*)

This article appears in the USENIX ;login: online magazine in 2023.

On September 30th, 2019, the IT Security team at the University of Chicago (*UChicago*) identified suspicious activity on over 100 user accounts. As a result, they locked those accounts and forced those users to change their password. As we explain below, it turns out that every affected UChicago account was reusing a password that had also been used on Chegg, an education technology company that provides homework help, textbook rentals, and other student services. Just a few weeks earlier, a data breach of Chegg had become public. Chegg relied on unsalted MD5—an outdated cryptographic hash function—to store the passwords of their 40 million subscribers. Furthermore, they did not have a process in place for deleting old accounts [4].

Faced with an ever-increasing number of accounts that require a password, many users cope by using similar passwords—or even the exact same password—across different accounts. Unfortunately for users, attackers leverage this propensity to compromise accounts. When any service suffers a data breach, attackers typically try to log into other services with the same email address or username alongside a password that is either the same as the leaked password or tweaked in simple ways. Even if a user follows every piece of advice about creating a strong password, all it takes is a single data breach to put them at risk if they have reused that password for any other account. Attackers' ability to conduct attacks exploiting reused passwords has increased as hundreds of websites have had their password databases stolen and leaked over the last decade [8].

This state of affairs raises many questions for system administrators and IT leaders trying to protect their organizations. How do threats from password reuse compare to threats from users choosing common, easy-to-guess passwords? For how long do accounts remain vulnerable? Out of hundreds of data breaches, how important is it to account for them all? How often do attackers appear to have exploited reused passwords, and what factors make them more likely to have done so?

To answer these sorts of questions, our team of academic researchers collaborated with our university's IT Security and Identity Management teams to conduct a twenty-year retrospective analysis of our university's vulnerability to password-guessing attacks. 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. As a result, the university's IT Security and Identity Management teams have a record of every password used by university affiliates over the past twenty years. 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. At the same time, because our analysis would enable us to identify potentially vulnerable accounts that had not yet been exploited, this collaboration enabled our university to better protect users against future attacks.

Our full study [10] appears at the USENIX Security Symposium in August 2023. In this article, we summarize our key findings, focusing on recommendations and lessons for organizations trying to defend against attacks exploiting password reuse.

Measurement Methodology

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 and academic records). A few years ago, the university began requiring current faculty, staff, and students to use Duo two-factor authentication (*2FA*).

We generated guesses for accounts by searching public data breaches. Breached credentials are commonly found as part of either a password database stolen from a single provider (e.g., LinkedIn), which we term an *individual service breach*, or a collection of hundreds of millions or even billions of credentials from many different services packaged into a *breach compilation* (e.g., Collection #1) [7].

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, we searched over 450 individual service breaches and 12 breach compilations for credentials potentially associated with those usernames. Specifically, we looked for passwords associated with a university email address (e.g., `blase@uchicago.edu`), having the same standalone username (e.g., `blase`), or having the same username as part of an email address with a different domain (e.g., `blase@gmail.com`). When we found hashes, rather than plaintext credentials, we attempted to crack them. We then used four state-of-the-art methods to tweak credentials (e.g., `monkey1` → `Monkey1!`). These modifications account for university affiliates using passwords that were similar to, but not exactly the same as, passwords used for other accounts. To compare the risk of password reuse to the risk of common passwords, we also used the LinkedIn individual service breach to generate thousands of the most common passwords to guess for all accounts, which included translating applicable common passwords to our context (e.g., `LinkedIn2012` → `UChicago2012`).

We, the computer science researchers, then sent our 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. For correct guesses, the IT Security team returned pseudonymous metadata for the guess (without the corresponding username and password) augmented with additional metadata (e.g., when the password was created, the user's current university affiliation). To protect users, the IT Security team 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, the IT Security team 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, these notifications described the research, explained the dangers of password reuse, and gave participants the opportunity to withdraw their data from the research.

Given the sensitivity of passwords and account security, our team carefully designed this research protocol collaboratively with numerous stakeholders at our university, including university leadership, the director of our institutional review board (IRB), the IT team, the alumni association, and more. Properly handling user data and minimizing risk to users were primary concerns. The experimental setup ensured that the IT Security team contacts were the only people that had access to the password history database and that the academic researchers never learned the identities of the users associated with correct guesses.

Key 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. While 12,247 correct guesses exploited reused passwords, only 1,979 exploited common passwords; 65 correct guesses fell into both categories. This finding underscores the far greater risk posed by attacks leveraging reused passwords even if (as we did) common passwords are customized for the attacked service.

Our guesses were most likely to be successful when we found credentials associated with a UChicago email address in a data breach, though our other potential matches also bootstrapped many correct guesses. Exploiting password reuse, we successfully guessed passwords for 32.0% of accounts we matched to a UChicago email address in a data breach, compared to 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.

For passwords created during approximately the first 12 years of the password history database, the number of accounts that were using a password that we eventually guessed increased every year, as seen in Figure 1. In late 2014, however, the number of accounts with passwords we could guess started to decrease. This drop coincides with a change to the password policy at the university. Instead of requiring that passwords be at least eight characters long and use three character classes (e.g., upper

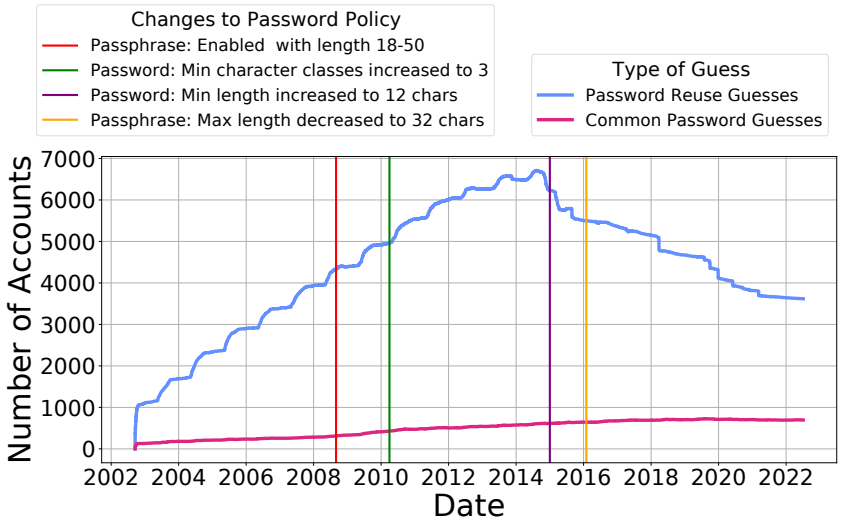

Figure 1: At the time indicated on the x-axis, the number of accounts that were actively using a password we correctly guessed in our study.

case letters or symbols), newly created passwords needed to be at least 12 characters long with the same character-class requirements. The university had changed its password policies before, but the effect was not as pronounced on the number of accounts with guessable passwords.

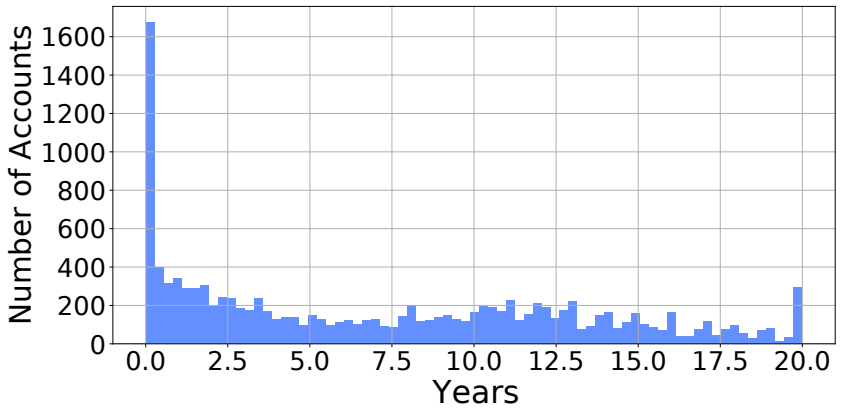

Figure 2: The length of time for which correctly guessed passwords (including those currently valid) had been active at the time they were changed, either by the user or as a result of a forced reset due to us guessing their password in this study.

Many accounts remained vulnerable for years. Of the passwords we guessed, the median time for which they remained valid was 6.2 years, with a maximum of 19.8 years. Figure 2 shows the full distribution of how long those passwords remained valid.

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. Table 1 shows some of the most effective data breaches and breach compilations in our study. Particular data breaches were more effective for different groups of users. For example, credentials from LinkedIn were particularly effective at guessing employees' passwords, and credentials from Chegg (the aforementioned homework-help site) at guessing students' passwords.

Figure 3 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. For example, five years after the LinkedIn breach was made public, roughly half of the 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 the peak in accounts using a related password.

Table 1: Individual service breaches (top) and breach compilations (bottom) that bootstrapped 25+ correct guesses. This table lists the reported date of the breach and the potential matches we identified (based on UChicago emails, usernames, or emails at other domains).

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

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

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

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 that we did not acquire. 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 Have I Been Pwned are typically employed when users create a password, so they would miss the (far more common) former cases.

Though 54.7% of correct guesses were based on verbatim reuse (exactly matching the breached password), the rest required password tweaking using previously published methods. The most successful password-tweaking strategies involved toggling the case of the first character (e.g., `monkey` → `Monkey`) or appending either "!" or "1" to the password. While we found that a recent deep-learning-based approach [11] produced the best ordered list of transformations "out of the box," the heuristics-based methods [3, 14, 12] we tested may have been more successful than the deep-learning-based approach had their guesses been reordered.

Our study also provided insight into the degree to which the vulnerabilities caused by password reuse are actually exploited. 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. We compared our correct guesses with apparent exploitation by attackers. 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 2 shows the six days on which over 100 accounts whose passwords we guessed exhibited suspicious activity. Some of this exploitation was quick. Returning to the story with which we began this article, all 134 accounts whose passwords we guessed that exhibited suspicious activity on September 30th, 2019, were found in the Chegg breach.

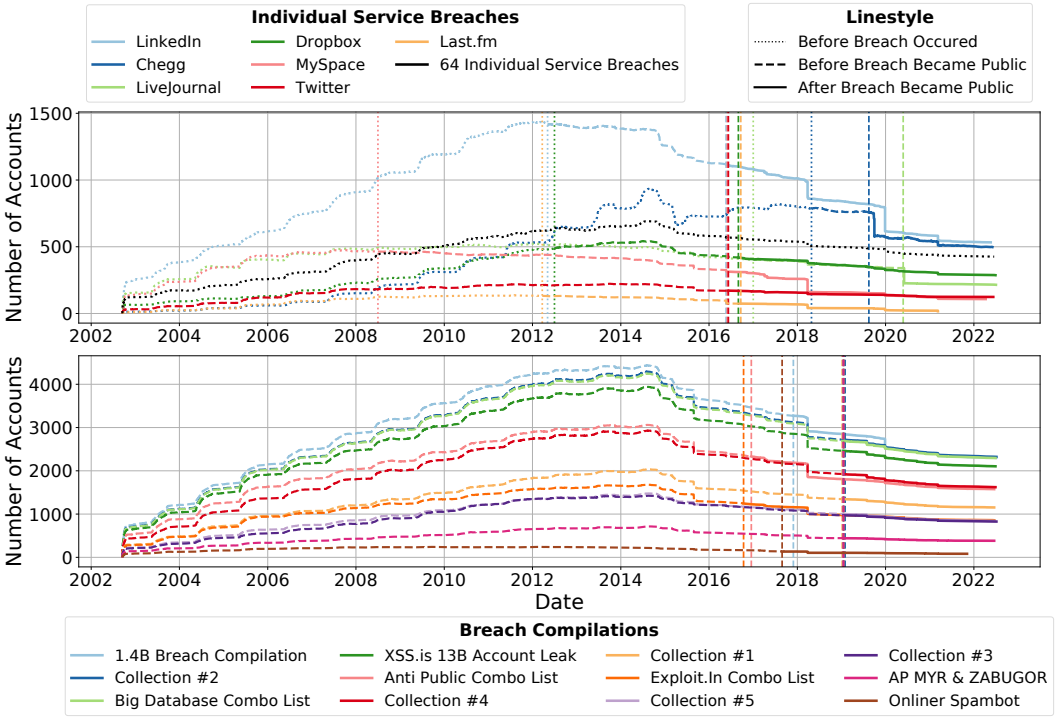

Figure 3: Number of accounts vulnerable over time from individual service breaches (top) and breach compilations (bottom).

Notably, that breach was added to Have I Been Pwned less than two months earlier. In contrast, other accounts were compromised years after their corresponding data breaches were posted publicly.

Table 2: The six days when over 100 accounts whose passwords we guessed exhibited suspicious activity and the associated breaches that bootstrapped the guesses.

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)

These apparent compromises were most likely for exact email matches (i.e., a `uchicago.edu` email address was found in the breach) and verbatim reuse (i.e., the UChicago password was exactly the same as the one found in the breach). Specifically, 60.7% of the apparent compromises in our dataset were an exact email match whose password was found verbatim in plaintext.

To better understand users' experiences, our university's IT team facilitated a survey of a sample of users whose passwords we guessed. Among the 40 respondents, most were unaware of the risks to their university account. Several were not even aware they had an account on the breached site. With users not knowing their accounts are at risk, it becomes even more important for system administrators to take steps that will mitigate these vulnerabilities.

Defending Against Attacks Leveraging Password Reuse

Protecting an organization from reused passwords is 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 [2].

In our study, we found that many UChicago accounts had reused passwords, and we saw evidence that some of these accounts had been previously compromised. We found many other accounts that were likely vulnerable, yet had not yet been compromised. This state of affairs presents latent risk for our university.

Based on our findings, we recommend that defenders take the following actions to proactively defend against attacks that leverage password reuse:

1. Proactively check for compromised credentials both at account creation and subsequently
2. Pay particular attention to organization-adjacent breaches (e.g., the Chegg and LinkedIn breaches for an educational institution)
3. Not ignore the long tail of individual service breaches
4. Check for similar email matches and username matches, not only exact email matches
5. Save computational resources by using heuristic tweaking algorithms, not those based on deep learning
6. Crack hashes to protect against motivated attackers
7. Implement processes to expire unused accounts
8. Help users understand the risks of password reuse
9. Consider moving away from user-chosen passwords entirely for online authentication

In recent years, researchers and practitioners have developed compromised-credential-checking tools. For instance, the Have I Been Pwned service [8] offers a Pwned Passwords API that many organizations use to check for compromised credentials. In fact, since 2019, our own university has used this API to check for compromised credentials when a user creates a password, as is typical. However, our study demonstrated that many vulnerable passwords were created at UChicago before a corresponding data breach was made public, or even before it occurred. This finding highlights the necessity of periodically checking existing credentials for compromises, which presents particular challenges for accounts that rarely log in.

In our study, vulnerable passwords came from an array of individual service breaches and breach compilations. High-profile leaks like LinkedIn enabled a significant number of correct guesses in our study. Further, we observed a high correlation with leaks from academic-related services like Chegg. There was a 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 such breaches may help stave off such rapid attacks. That said, skipping over smaller individual service data breaches or large (poorly formatted) compilations may cause defenders to miss at-risk accounts. Unfortunately, processing breaches requires time investments from defenders.

From our study, we learned that adequately protecting user accounts requires accounting for multiple ways of matching users, tweaked passwords, and credentials that are only publicly available in hashed form. While exact email matches (`@*.uchicago.edu`) accounted for 4,585 users' vulnerable accounts in our study, we matched an additional 6,951 users' vulnerable accounts via email matches from non-UChicago domains, which differs from how Have I Been Pwned is often used and how prior work has typically investigated password reuse. Checking for password reuse with only exact email matches may not be enough to protect users from motivated attackers. Furthermore, users seem to reuse passwords verbatim more often than they marginally tweak passwords. While 55% of correct guesses exactly matched the original password found in a data breach, the remaining 45% required transformations. One of the most computationally intensive steps in our study was using a recent deep learning approach to tweak guesses [11]. While it produced the best ordered list of transformations "out of the box," heuristics-based methods [3, 14, 12] are likely a better tool for defenders who are not trying to optimize each guess. They produced most of the same tweaked passwords as the deep learning approach, as well as others the deep learning approach did not, at a much lower computational cost. In the same vein, 14.7% of our successful guesses were found only as hashes. Diligent defenders might consider trying to crack these hashes as part of proactive credential checking.

Furthermore, we found that reused passwords are at risk for long periods of time. Many accounts remained vulnerable for years, including as student accounts transitioned to alumni accounts. Some were exploited years after the corresponding data breach. Many organizations currently do not expire passwords [5], but perhaps expiration over long periods or inactive accounts should be considered.

More work into securing legacy accounts is necessary from the research community.

We also found through our small-scale survey that many users were unaware that their UChicago accounts were at risk due to password reuse. Some users were not even aware that they had accounts on the corresponding breached sites to begin with. These shortcomings highlight the need for improving communication with users. Chrome [13], Firefox, and Safari notify users if their passwords appear in a data breach. The Have I Been Pwned service, itself integrated with password managers, enables users to check for their appearance in a data breach. Supporting these efforts, academic work has sought to improve the usability of data breach notifications [6], but more work is needed.

Most crucially, however, the complex vulnerabilities caused by password reuse highlight the huge shortcomings of relying on user-chosen passwords as a single line of defense for authentication. A few years ago, UChicago began requiring current faculty, staff, and students to use Duo 2FA. Thus, an attacker successfully guessing a password is now insufficient for compromising an account. While adoption of voluntary 2FA is often low, prior work at another university found acceptable user buy-in when 2FA was mandated [1], as it is at UChicago. Thus, organizations that continue to rely on passwords should require 2FA for all accounts and engage in proactive credential checking as described above.

Currently, though, the community seems to be at an inflection point in the transition to passwordless authentication. For instance, the FIDO2 standard enables users to authenticate on the web using public key cryptography combined with local authentication, such as with a PIN or fingerprint, and these schemes are becoming increasingly usable [9]. Ideally, more organizations would move away from using passwords as the primary method of authentication, or even as a fallback method. Indeed, our university is planning to make the switch to passwordless authentication in the near future.

References

[1] Jessica Colnago, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Lorrie Cranor, and Nicolas Christin. "It's Not Actually That Horrible": Exploring Adoption of Two-Factor Authentication at a University. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, Montréal, Québec, Canada, April 2018. ACM.

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

[3] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The Tangled Web of Password Reuse. In *Proceedings of the Symposium on Network and Distributed System Security*, NDSS '14, San Diego, California, USA, February 2014. ISOC.

[4] Federal Trade Commission. FTC Brings Action Against Ed Tech Provider Chegg for Careless Security that Exposed Personal Data of Millions of Customers, October 2022. https://www.ftc.gov/news-events/news/press-releases/2022/10/ftc-brings-action-against-ed-tech-provider-chegg-careless-security-exposed-personal-data-millions

[5] Eva Gerlitz, Maximilian Häring, and Matthew Smith. Please Do Not Use !?_ or Your License Plate Number: Analyzing Password Policies in German Companies. In *Proceedings of the Symposium on Usable Privacy and Security*, SOUPS '21, pages 17–36, Virtual Conference, August 2021. USENIX.

[6] 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 *Proceedings of the ACM Conference on Computer and Communications Security*, CCS '18, pages 1549–1566, Toronto, Ontario, Canada, October 2018. ACM.

[7] Andy Greenberg. Hackers Are Passing around a Megaleak of 2.2 Billion Records. *Wired*, January 2019. https://www.wired.com/story/collection-leak-usernames-passwords-billions/

[8] Troy Hunt. Have I Been Pwned? – Pwned Websites, July 2023. https://haveibeenpwned.com/PwnedWebsites

[9] Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. "It's Stored, Hopefully, on an Encrypted Server": Mitigating Users' Misconceptions About FIDO2 Biometric WebAuthn. In *Proceedings of the USENIX Security Symposium*, SSYM '21, pages 91–108, Virtual Conference, August 2021. USENIX.

[10] 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. In *Proceedings of the USENIX Security Symposium*, SSYM '23, Anaheim, California, USA, August 2023. USENIX.

[11] Bijeeta Pal, Tal Daniel, Rahul Chatterjee, and Thomas Ristenpart. Beyond Credential Stuffing: Password Similarity Models using Neural Networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, SP '19, pages 866–883, San Francisco, California, USA, May 2019. IEEE.

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

[13] 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 *Proceedings of the USENIX Security Symposium*, SSYM '19, pages 1556–1571, Santa Clara, California, USA, August 2019. USENIX.

[14] 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 *Proceedings of the ACM Conference on Data and Application Security and Privacy*, CODASPY '18, pages 196–203, Tempe, Arizona, USA, March 2018. ACM.