Post Breach Password Audit — Case Study

Prepared by Mohamed Elsaban
Date: August 20, 2025

Executive Summary

This report describes a practical password security audit that I performed after discovering that a password I once used appeared in a published breach dataset. The goal was to understand real world risk from exposed credentials and to practice responsible verification techniques. I used only open sources and paid intelligence portals that lawfully index breach data. For any hands on verification I obtained explicit consent from the account owners. No unauthorized access was attempted and no plaintext credentials were stored. The audit confirmed that while bcrypt with a cost parameter of ten resists brute force, personalized wordlists can still recover weak choices when users reuse predictable patterns. The outcome is a set of practical recommendations for stronger authentication hygiene.

Scope and Ethics

- Focus on learning and defense. The objective was to measure risk and improve personal and team practices.
- Use of breach intelligence sources that aggregate already public breach data. No intrusion into systems was performed.
- Consent obtained in writing from a small group of owners whose records appeared in the dataset. Work limited to those records.
- No storage of plaintext credentials. Any verified secrets were rotated immediately by the owners.
- All artifacts and partial datasets were deleted after the audit concluded.

Background and Discovery

I first checked my own exposure using a public service that reports known breaches. An email check showed no recent exposure but a password check flagged a match associated with a transportation company breach from the year twenty twenty. I reviewed public reporting to understand the scope and decided to study the problem further using open source intelligence resources.

Data Sourcing

I surveyed several breach intelligence portals and discussion spaces to identify sources that index the dataset. One commercial portal quoted a very high access fee which was not appropriate for a student research effort. I then located a lawful subscription portal that offered weekly access at a modest cost and used it to confirm the presence of records. The fields available included names, phone numbers, email addresses and hashed passwords. Password hashes were stored using bcrypt with a cost parameter of ten.

Methodology

The audit combined non intrusive analysis with a consent based verification step. The steps below summarize the approach in plain language.

1. Identify algorithm and parameters. Verified that the hash format matched bcrypt with cost ten. Brute force was considered infeasible for generic recovery.
2. Select consented samples. Three volunteer records were selected from the dataset. No hints were accepted from the owners to keep the test realistic.
3. Build personalized wordlists. Used a tool known as CUPP which stands for Common User Passwords Profiler to generate candidate passwords from biographic cues and likely variations. Input data was limited to non sensitive facts and simple guesses.
4. Verify against hashes. Used a standard offline password recovery tool on a personal computer with a graphics card to test the wordlists against the consented hashes.
5. Record outcomes and rotate credentials. Any recovered password was documented privately and immediately changed by the owner. No further use was made of the recovered values.

Findings

- Two of the three consented hashes were recovered quickly with targeted wordlists of approximately one hundred eighty thousand candidates each. The third remained unrecovered within the defined effort limit.
- The recovered passwords followed predictable themes such as nicknames plus dates and simple substitutions. These patterns remain common even among technical users.
- Bcrypt with cost ten effectively prevents naive brute force in a reasonable time frame for a single consumer graphics card. Targeted guessing is the main practical risk when users rely on patterns drawn from their public presence.
- The total compute time for each successful recovery was on the order of minutes which aligns with a realistic threat actor using focused dictionaries rather than massive brute force resources.

Risk Assessment

When a password appears in a breach dataset the primary risk is reuse across multiple services. Even when a hash is strong the presence of the hash confirms that the string has been exposed to adversaries for analysis. If the exposed string follows predictable patterns it may be recovered with modest effort. Public footprints on social media and casual biographies can provide enough material to build effective wordlists. The most significant organizational risk comes from reused passwords on email or single sign on and from the lack of multifactor authentication.

Recommendations

- Adopt a password manager and move to unique random passwords for every service.
- Enable multifactor authentication for all important accounts with a preference for authenticator apps or security keys.
- Monitor exposure using public services that report known breaches and enroll in organization wide notifications where available.
- Educate users about predictable patterns such as names, dates and keyboard sequences and why these are risky even with strong hashing.
- For administrators, consider a confidential process to help staff rotate exposed credentials and to validate that reuse has been eliminated.
- For security teams, maintain access to a lawful breach intelligence service and use it to inform targeted outreach rather than public disclosure.

Outcome and Learning

The audit took about two months part time while I was studying and working. It improved my understanding of practical password risk and the value of consent based testing. It also strengthened my open source intelligence skills in responsible ways. Most importantly it led to better personal hygiene including a password manager and consistent multifactor authentication. I am prepared to repeat this process for an organization with formal approvals and with clear legal and privacy guardrails.

Appendix: Practical Notes

- CUPP stands for Common User Passwords Profiler. It can generate focused wordlists using simple facts and likely variations.
- Hash verification was performed on a home computer with a consumer graphics card. Cloud resources were not required for the limited scope of this test.
- No commands are included in this report to keep the focus on outcomes and ethics. The methods are standard and well documented in public sources.