In Situ Key-Blocking using Trajectory Constraints to Avoid Reuse of Passwords

Sreecharan Bojja[1 [0009−0004−7552−9651]], Vijayanand Banahatti[1 [0000−0003−0321−498X]], and Sachin Lodha[1 [0000−0001−5771−4977]]

TCS Research, Tata Consultancy Services, Pune, India
{b.sreecharan,vijayanand.banahatti,sachin.lodha}@tcs.com

Abstract. Passwords remain widely deployed despite the increasing availability of biometrics and multi-factor authentication. As a result, password guessing, credential stuffing, and password reuse attacks continue to pose significant security risks [5, 10]. Existing password defenses are largely reactive: they rely on post-hoc rejection, blacklist checks, or strength indicators after users have already moved toward potentially insecure choices [2, 7]. We introduce a proactive password-creation mechanism that evaluates risk during password entry and intervenes before insecure passwords are completed. Our approach models password creation as a trajectory through password space and combines reuse-related risk derived from neighborhood density within leaked-password corpora with predictability estimates from character-level models. We implement this idea in a virtual keyboard that temporarily blocks next-character actions predicted to substantially increase risk while preserving user agency in real time. Through empirical analysis and offline simulations, we show that password-space structure supports interaction-time risk assessment, that proactive intervention occurs earlier than post-hoc rejection, and that generated passwords shift away from higher-risk regions. These results suggest that interaction-time trajectory control is a promising direction for human-centered password security.

Keywords: Password Reuse · Password Predictability · Security Intervention

1 Introduction

Passwords remain widely used because they are familiar, easy to deploy, and supported across online services. However, password-based authentication remains vulnerable to password guessing, credential stuffing, and reuse-based attacks. Prior work has shown that users often create passwords that are predictable, reused across services, or structurally similar to leaked credentials [4, 5, 10], making manual password creation a persistent human-centered security problem. Existing defenses include password-composition policies, compromised-password blacklists, probabilistic strength estimators, and real-time strength meters [1, 2, 7, 9]. Although these mechanisms improve password evaluation, they are largely

reactive. Blacklists and composition policies typically reject passwords after completion, while strength meters provide advisory feedback [3, 7] that users must interpret and act upon. As a result, users may continue along risky creation paths and revise passwords only after weakness has already emerged. We argue that password weakness is not only a property of completed passwords, but also emerges progressively during password creation. Early character selections can steer users toward regions of password space associated with high reuse, low entropy, or similarity to compromised credentials. This motivates a trajectory-based view of password creation, where risk is assessed over partial inputs rather than only after completion. In this paper, we introduce a proactive password-creation mechanism that evaluates risk during entry and temporarily blocks next-character actions predicted to substantially increase the risk of the evolving password trajectory. Our approach combines reuse-related risk, estimated from neighborhood density within leaked-password corpora, with predictability risk derived from character-level probabilistic models. The central contribution is not a new password-strength estimator, but a shift in the point of intervention from post-hoc evaluation and advisory feedback to interaction-time action-space constraint. Our contributions are:

- We model password creation as a trajectory that enables continuous risk assessment over partial inputs.
- We combine reuse-related and predictability-based signals to estimate trajectory risk.
- We propose a proactive intervention mechanism that temporarily blocks locally high-risk next-character actions.
- Through empirical analysis and offline simulations, we demonstrate the feasibility of interaction-time password intervention to anticipate risky trajectories and guide users toward lower-risk password choices.

2 Related Work

Prior work has proposed password-composition policies, compromised-password blacklists, probabilistic strength estimators, and password-strength meters [1, 2, 7, 9] to reduce weak password choices. These techniques improve password quality by rejecting known weak passwords or communicating estimated strength, but they largely remain reactive: users are evaluated after entering potentially insecure sequences and must revise passwords through repeated attempts. Real-time feedback systems, including strength meters, warnings, nudges, and recommendation interfaces, improve user awareness during password creation but generally remain advisory [3, 7]. Users must interpret feedback and decide how to modify their passwords. Our approach differs by intervening directly in the action space: next-character choices predicted to substantially increase risk are temporarily constrained before insecure passwords are completed. Password reuse and similarity are central to credential-stuffing and account-compromise attacks [5, 10]. Software-based keyboards have also been studied in password-entry

contexts, particularly for understanding usability issues during password creation and entry [8]. Users often reuse passwords across services or create variants of previous credentials, allowing attackers to exploit leaked password datasets beyond exact matches. Building on this observation, our work treats password weakness as a trajectory through password space and investigates whether reuse-related and predictability-based risk can be assessed continuously during password creation.

3 Methodology

3.1 Password Creation as a Trajectory

Password creation can be modeled as a sequential process in which users append characters to an evolving prefix [1, 3]. Each partial prefix occupies a position in password space, and each subsequent character selection influences the set of possible completions. This motivates a trajectory-based view of password creation, where risk is assessed continuously over partial inputs rather than only after a password is completed. Our goal is to determine whether risk can be estimated during password creation and whether intervention can occur before insecure passwords are finalized. We estimate trajectory risk using two complementary signals:

- **Reuse-related risk**, estimated from neighborhood density within leaked-password corpora.
- **Predictability risk**, estimated using character-level probabilistic models.

These signals capture different dimensions of password weakness. Reuse-related risk reflects whether a partial password is moving toward regions that resemble known leaked or frequently occurring passwords, while predictability risk reflects whether the evolving sequence follows common character-level patterns that may be easier for attackers to guess.

3.2 Reuse-Related Density

We use leaked-password datasets as a reference corpus for estimating reuse-related risk. The goal is to determine whether an evolving password lies near regions heavily represented in compromised credential data. Unlike exact-match blacklists, this allows the model to capture small variations of leaked or commonly used passwords. Passwords are represented using character-level 3-gram features. We transform the 3-gram counts using term frequency-inverse document frequency, or TF-IDF, weighting. Because the resulting vectors are high-dimensional and sparse, we apply truncated singular value decomposition to reduce each password representation to 50 components. The reduced vectors are subsequently L2-normalized. For each password representation, we identify its $k = 20$ nearest neighbors in the reduced feature space, using cosine distance as distance metric. Local neighborhood density is then estimated from the resulting nearest-neighbor distances. Passwords with closer neighbors are interpreted

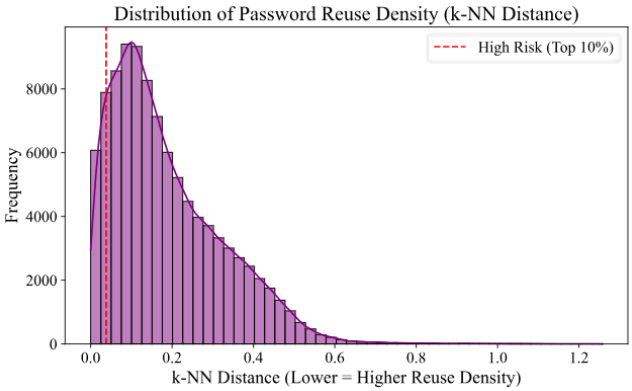


Fig. 1: Distribution of password reuse-related density measured using k-nearest-neighbor distance. Lower k-NN distance indicates denser regions of password space.

as belonging to denser leaked-password neighborhoods, whereas passwords with more distant neighbors are interpreted as more structurally isolated. Unlike an exact blacklist, the neighborhood representation allows the signal to capture structurally related password variants rather than exact matches alone.

3.3 Predictability Risk

Password risk is also influenced by structural predictability. A password may be uncommon in a leaked-password corpus but still follow a highly predictable pattern, such as a keyboard sequence, dictionary-like prefix, or common character substitution. Therefore, reuse-related density alone is insufficient for estimating password weakness. To capture predictability, we use character-level probabilistic models with a maximum context length of four characters that estimate the likelihood of a password prefix under common password-generation patterns [1, 9]. Prefixes that are more likely under the model are assigned higher predictability risk. This score is normalized so that larger values consistently indicate higher risk. Combining reuse-related and predictability-based signals allows the system to capture two complementary attacker advantages: similarity to leaked-password neighborhoods and structural ease of guessing.

3.4 Risk Scoring

For a password prefix p_t, we define the overall risk score as a weighted combination of reuse-related risk and predictability risk:

$$R(p_t) = \alpha R_{reuse}(p_t) + (1 - \alpha) R_{pred}(p_t) \tag{1}$$

Table 1: Effect of the intervention threshold on model-defined risk and key availability over simulated trajectories.

Threshold T	**Mean Risk**	**Avg. Keys Disabled**
40	18.95	6.66%
60	33.75	1.51%
80	**55.30**	**0.24%**
100	64.82	0.00%

where $R_{reuse}(p_t)$ denotes normalized reuse-related risk derived from neighborhood density within the leaked-password corpus, and $R_{pred}(p_t)$ denotes normalized predictability risk derived from the character-level probabilistic model. In our experiments, we set $\alpha = 0.5$ to weight both signals equally. The risk score is not intended to provide an absolute measure of password security. Rather, it provides a comparative signal for evaluating whether a candidate next-character action increases or decreases the risk of the current password trajectory.

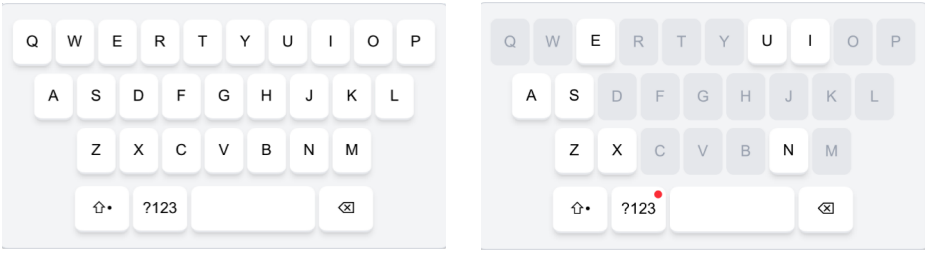
3.5 Proactive Intervention by Targeted Temporary Blocking

At each step of password creation, the system evaluates the risk associated with each possible next-character action. For a candidate character c, the system computes the resulting change in risk:

$$\Delta R_t(c) = R(p_t + c) - R(p_t) \tag{2}$$

If $\Delta R_t(c)$ exceeds a predefined threshold T, the character c is temporarily blocked for the current prefix. Blocking is local and reversible: it applies only to the immediate next action and is recomputed as the password prefix changes. Users can continue password creation by selecting from the remaining available characters. This mechanism differs from conventional password-strength meters. Strength meters estimate password quality and present advisory feedback, but users remain responsible for interpreting the feedback and revising their passwords. Our approach instead intervenes directly in the action space by constraining next-character actions predicted to substantially increase risk. The contribution is therefore not a new strength-estimation model, but a shift in the point of intervention from post-hoc evaluation and advisory feedback to interaction-time trajectory control.

In our experiments, we use a fixed threshold of $T = 80$, selected through a preliminary sensitivity analysis of $N = 200$ simulated password-creation trajectories as shown in Table 1. At this threshold, an average of only 0.24% of candidate keys were disabled, satisfying our design criterion of keeping the blocking rate below 1% to preserve a large available action space. The threshold controls the aggressiveness of intervention and may be calibrated in future deployments according to the desired trade-off between model-defined risk reduction and user flexibility.



(a) Initial state (b) State for the prefix "mas"

Fig. 2: Prototype virtual keyboard showing the initial state and a blocked-key state for prefix “mas”

4 Evaluation and Results

Our evaluation examines whether trajectory-based risk estimation can support proactive intervention during password creation. Since our goal is to study the feasibility of interaction-time intervention, we evaluate the proposed mechanism using empirical analysis of leaked-password datasets [6] and offline password-creation simulations. Across all experiments, we use the same risk formulation and fixed intervention threshold to ensure that observed differences arise from the intervention strategy. We compare no intervention, post-hoc rejection, and proactive intervention over $N = 1,000$ trajectories generated by template-based and Markov-based simulated typists. No intervention applies no constraints; post-hoc rejection discards passwords with zxcvbn scores of atleast 2, allowing up to 10 attempts; and proactive intervention temporarily blocks candidate characters satisfying the intervention criterion.

4.1 Validating the Risk Signals

We first evaluate whether neighborhood density within leaked-password corpora can support reuse-related risk estimation. To examine this assumption, we represent passwords using character-level n-gram features and compute k-nearest-neighbor distances within the leaked-password corpus. Lower k-NN distances correspond to denser regions of password space, while higher distances correspond to sparser regions. To validate the relationship between density and reuse-related risk, we compare neighborhood density against observed password frequency in the leaked-password data. We observe a moderate positive relationship between density and frequency (Pearson’s $r = 0.314$, $p < .001$; Spearman’s $\rho = 0.318$, $p < .001$). This suggests that passwords in denser regions tend to occur more frequently in leaked datasets, supporting the use of neighborhood density as a proxy for reuse-related risk. We further compare dense and sparse password neighborhoods using average Levenshtein edit distance between passwords and their nearest neighbors. As shown in Table 2, passwords in dense regions exhibit much lower neighborhood edit distances than passwords in sparse regions. This

Table 2: Structural Tightness of Dense and Sparse Password Neighborhoods

Password	k-NN	Type	Avg. ED	Representative Neighbors
1234567b	0.00	Dense	1.20	1234567B, 1234567C, 1234567S
password1	0.00		2.75	pAssword1, PassWord1, PASSword1
iloveyou1	0.00		2.80	iLOVEyou1, iloveyoU1, ILoveYou1
jramm13	0.63	Sparse	10.00	tummytree27, tummytree24, tummytree78
lexinantara	0.62		9.45	fainalfantasy, myfinalfantasy, saldinata
10RENN	0.63		8.10	rennhok10, irene0175, reneysus010663

indicates that dense regions contain structurally similar password variants, while sparse regions contain more isolated strings. These results suggest that steering users away from dense regions may reduce movement toward common variants of weak or leaked passwords. We also analyze whether reuse-related risk and predictability risk capture distinct aspects of password weakness. The correlation between the two signals is weak (Pearson's $r = 0.15$, $p < .005$; Spearman's $\rho = 0.14$, $p < .005$), suggesting that the signals are related but not redundant. This supports combining reuse-related density with character-level predictability in the overall trajectory-risk score.

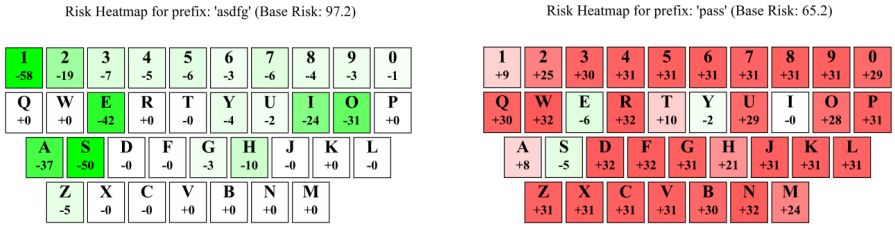
4.2 Risk Emergence During Password Creation

We next examine whether password risk can be identified before password completion. For each prefix, the system evaluates the risk associated with possible next-character actions. Figure 3 shows keyboard heatmaps for example prefixes such as "asdfg" and "pass". The heatmaps show that risk is not uniformly distributed across the action space. Instead, only a subset of next-character choices substantially increases estimated trajectory risk, while many alternative continuations remain available. This supports the central motivation for proactive intervention: password weakness can begin to emerge during password creation, before the final password is submitted.

Therefore, intervention can occur at the level of local next-character actions rather than only after password completion. To further evaluate whether partial prefixes provide information about final password risk, we simulate password-creation trajectories (with a max length of 12 characters) and compute the correlation between prefix risk and final password risk at different prefix lengths. The correlation is modest at 3 characters ($r = 0.144$), increases at 5 characters ($r = 0.218$), and becomes substantially stronger at 8 characters ($r = 0.592$). These results suggest that risk becomes increasingly predictive before password completion, particularly at mid-length prefixes.

4.3 Intervention Behavior and Final Risk

We next evaluate whether proactive blocking changes both the timing and outcomes of simulated password creation. Across the simulated trajectories, intervention remains localized: blocking occurs only at selected keystroke positions, and only a small fraction of the available action space is disabled at any step,



(a) Prefix "asdfg" (b) Prefix "pass"

Fig. 3: Qwerty heatmaps visualizing the real-time contribution of certain characters to the overall risk score. Lighter shade indicates little contribution to overall risk score variation and darker shade indicate high contribution to the overall risk score.

Table 3: Step-by-Step Intervention Rates During Password Entry

Keystroke Position	**Active Blocking Probability (%)**	**Keyboard Keys Disabled (Avg. %)**
4	8.00	0.11
5	26.50	0.38
6	15.50	0.22
7	3.00	0.04
8	6.00	0.14

as shown in Table 3. These results indicate that the mechanism targets selected high-risk continuations while preserving most candidate actions. Because the evaluation is simulation-based, the intervention rate should be interpreted as a measure of action-space constraint rather than as direct evidence of perceived user agency.

We then compare password outcomes under three policy scenarios: no intervention (baseline), post-hoc rejection, and proactive intervention. Table 4 reports the mean and maximum final model-defined risk, the percentage reduction in mean final risk relative to the baseline, the mean estimated $\log_{10}$ guesses, and the mean blind entropy across $N = 1{,}000$ simulated password-creation trajectories. The baseline condition produces a mean final risk of 53.93 and a maximum observed final risk of 83.47. Post-hoc rejection produces similar outcomes, with a slightly higher mean final risk of 54.29 and a maximum observed final risk of 83.52. This corresponds to a -0.67% mean-risk reduction relative to the baseline, indicating that post-hoc rejection does not reduce the model-defined risk under the evaluated simulation protocol. Its mean estimated $\log_{10}$ guesses increases marginally from 7.45 to 7.49, while its mean blind entropy increases from 54.93 to 55.05 bits. Proactive intervention produces the strongest outcomes across the reported metrics. It reduces the mean final risk from 53.93 to 45.39, corresponding to a 15.83% reduction relative to the baseline, and low-

Table 4: Comparison of password outcomes across policy scenarios.

Policy	**Mean Final Risk**	**Max. Final Risk**	**Mean-Risk Reduction**	**Mean $\log_{10}$ Guesses**	**Mean Blind Entropy (bits)**
Baseline	53.93	83.47	0.00% (Reference)	7.45	54.93
Post-hoc	54.29	83.52	−0.67%	7.49	55.05
Proactive	**45.39**	**79.74**	**15.83%**	**8.66**	**55.77**

Table 5: Latency of full-keyboard risk evaluation across 500 updates. Each update evaluates all 70 candidate actions.

Runtime Metric	**Observed Value**
Median latency	75.51 ms
95th-percentile latency	80.39 ms
99th-percentile latency	88.13 ms
Maximum latency	117.65 ms
Memory footprint	∼6 GB

ers the maximum observed final risk from 83.47 to 79.74. It also achieves the highest mean estimated $\log_{10}$ guesses, increasing from 7.45 under the baseline to 8.66, and the highest mean blind entropy, increasing from 54.93 to 55.77 bits. Within the evaluated simulations, proactive intervention therefore reduces both mean and maximum model-defined risk while increasing the estimated guessing effort and blind entropy of the resulting passwords. Unlike post-hoc rejection, which evaluates a password only after completion, proactive intervention acts during password creation by restricting selected next-character continuations. These results provide evidence that interaction-time intervention can improve the evaluated password outcomes under the present simulation protocol. However, the reported maximum risks are finite-sample observations rather than formal safety bounds, and the results should not be interpreted as establishing universal superiority against all password-guessing strategies.

4.4 Interaction-Time Latency and Memory Utilization

To assess whether the proposed mechanism can evaluate candidate actions during password entry, we measured the latency of a complete keyboard update. Each update scores all 70 candidate next-character actions using the neighborhood-density and Markov-risk components. We evaluated 500 full-keyboard updates and recorded the median, 95th-percentile, 99th-percentile, and maximum latency. As shown in Table 5, the median, 95th-percentile, and 99th-percentile update latencies remain below 100 ms, while the maximum observed latency is 117.65 ms. These measurements indicate that the prototype can recompute the risk of all 70 candidate actions within approximately 76 ms for a typical update under the evaluated configuration. The current implementation requires approximately 6 GB of memory, primarily due to the FAISS nearest-neighbor index.

These results provide initial evidence that full-keyboard risk evaluation is computationally feasible during manual password creation. However, they should not be interpreted as a complete production-deployment evaluation. The memory footprint remains substantial, and latency may vary across hardware and reference-corpus configurations. Future implementations could investigate index compression, corpus reduction, approximate-search configurations, or model distillation to reduce memory and computational requirements.

5 Limitations and Future Work

Our evaluation relies on offline simulations and leaked-password data and therefore does not fully capture how users may adapt to temporary key blocking. Real users may select alternative characters, revise earlier input, restart password creation, or abandon the interaction. Consequently, the percentage of disabled keys measures action-space constraint, but does not directly establish perceived agency, usability, frustration, or memorability. A controlled user study is required because password-creation behavior is influenced by how guidance and feedback are presented [2, 7], and could build on prior evaluation of software-based keyboards for password creation [8]. The post-hoc baseline is also affected by its retry and fallback rules. Passwords with a zxcvbn score of at most 2 are regenerated for up to 10 attempts, after which the final password is retained even if it remains predictable or has high model-defined risk. The reported post-hoc outcomes therefore do not represent every possible rejection mechanism. Future experiments should report rejection rates, attempt counts, and retry-limit failures, evaluate alternative fallback rules, and allow participants to respond naturally to both key blocking and post-completion rejection. Because the proposed risk function determines which characters are blocked, reductions in this score demonstrate alignment with the intervention objective rather than independent resistance to guessing attacks. Estimated $\log_{10}$ guesses and blind entropy broaden the evaluation, but do not cover the Markov, neural, and rule-based attacker strategies considered in prior password research [1, 5, 9]. Future work should evaluate success at fixed guessing budgets using held-out attack models. Leaked-corpus neighborhood density is similarly a proxy for structural similarity, not direct evidence of account-level or cross-service reuse, which requires user-linked observations [10]. The evaluation also lacks comparisons with exact blacklist matching, edit-distance baselines, and existing password-similarity methods [2, 5]. Such comparisons are needed to establish whether the proposed TF–IDF, SVD, and nearest-neighbor representation improves upon simpler alternatives. Finally, our principal contribution is the interaction-time intervention mechanism rather than a fundamentally new password-strength estimator. Without comparative user evidence, we cannot conclude that localized key blocking is more usable or effective than strength meters, warnings, or post-completion rejection [1, 7, 9]. The favorable proactive results should therefore be interpreted as feasibility evidence under the present simulation assumptions, not as a gold standard, formal security guarantee, or proof of universal superiority. A complete

assessment requires stronger baselines, independent cracking experiments, and a controlled user study followed by security analysis of the passwords created under each condition.

6 Conclusion

We presented a proactive password-creation mechanism that models password entry as a trajectory through password space and intervenes during creation rather than after completion. By combining reuse-related density signals from leaked-password corpora with predictability-based risk estimation, our approach identifies locally high-risk next-character actions and temporarily constrains them while preserving alternative choices. Through empirical analysis and offline simulations, we show that interaction-time intervention can occur earlier than post-hoc rejection and can shift password creation away from higher-risk regions. These findings suggest that proactive trajectory control is a promising direction for human-centered password security, particularly in contexts where users continue to create passwords manually. Rather than replacing existing password-strength mechanisms, our work reframes password defense as an interaction-time intervention problem aimed at reducing movement toward predictable, reused, or compromised-password-like trajectories.

A Reproducibility and Supporting Artifacts

A.1 Reference Corpus and Preprocessing

We use RockYou2024 as a reference corpus for estimating leaked-password neighborhood structure. This signal is a proxy for structural similarity to compromised-password regions and does not directly measure account-level or cross-service password reuse. Passwords are represented using character-level 3-grams with TF-IDF weighting, reduced to 50 components using truncated SVD, and L2-normalized. Neighborhood density is then estimated using the 20 nearest neighbors. Because the reference corpus contains sensitive leaked-password data, we do not release the raw passwords or reversible representations of the corpus. Researchers reproducing the corpus-dependent analysis must obtain the reference data independently and use it in accordance with applicable institutional, ethical, and legal requirements.

A.2 Simulation Data and Replication Code

The replication code and simulation outputs are available at:

https://github.com/txmaniac/trajectory-constrained-password-creation-results

The artifact includes CSV files containing synthetic passwords generated by TemplateTypist and MarkovTypist under the baseline, post-hoc, and proactive conditions, together with the model-defined risk, estimated $\log_{10}$ guesses, blind entropy, policy labels, and other fields used in the reported analysis.

References

1. Castelluccia, C., Dürmuth, M., Perito, D.: Adaptive password-strength meters from markov models. In: Proceedings of the Network and Distributed System Security Symposium. NDSS '12, Internet Society (2012)
2. Habib, H., Colnago, J., Melicher, W., Ur, B., Segreti, S., Bauer, L., Christin, N., Cranor, L.F.: Password creation in the presence of blacklists. In: Proceedings of the 2017 Workshop on Usable Security. USEC '17, Internet Society (2017). https://doi.org/10.14722/usec.2017.23043
3. Komanduri, S., Shay, R., Cranor, L.F., Herley, C., Schechter, S.: Telepathwords: Preventing weak passwords by reading users' minds. In: 23rd USENIX Security Symposium. pp. 591–606. USENIX Security '14, USENIX Association (2014), https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/komanduri
4. Mutalik, R., Chheda, D., Shaikh, Z., Toradmalle, D.: Modeling human tendencies for password guessing. In: Tuba, M., Akashe, S., Joshi, A. (eds.) ICT Systems and Sustainability, Lecture Notes in Networks and Systems, vol. 321. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-5987-4_26
5. Pal, B., Daniel, T., Chatterjee, R., Ristenpart, T.: Beyond credential stuffing: Password similarity models using neural networks. In: 2019 IEEE Symposium on Security and Privacy. pp. 417–434. SP '19, IEEE (2019). https://doi.org/10.1109/SP.2019.00056
6. Petkauskas, V.: Rockyou2024: 10 billion passwords leaked in the largest compilation of all time. Cybernews (2024), https://cybernews.com/security/rockyou2024-largest-password-compilation-leak/, accessed: January 23, 2026
7. Shay, R., Bauer, L., Christin, N., Cranor, L.F., Forget, A., Komanduri, S., Mazurek, M.L., Melicher, W., Segreti, S.M., Ur, B.: A spoonful of sugar? the impact of guidance and feedback on password-creation behavior. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 2903–2912. CHI '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2702123.2702586
8. Shukla, M., Bojja, S., Jayakrishnan, G., Banahatti, V., Lodha, S.: Using graph analysis for evaluating usability of software-based keyboard for password creation. In: Indian Conference on Human-Computer Interaction Design and Research. pp. 215–239. Springer (2024)
9. Thai, B.L.T., Tanaka, H.: A study on markov-based password strength meters. IEEE Access **12**, 69066–69075 (2024). https://doi.org/10.1109/ACCESS.2024.3401195
10. Wang, C., Jan, S.T.K., Hu, H., Bossart, D., Wang, G.: The next domino to fall: Empirical analysis of user passwords across online services. In: Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy. pp. 196–203. CODASPY '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3176258.3176332