Leveraging LLMs to protect users against target password guessing

Manaswi Raj*
manaswiraj195@kgpian.iitkgp.ac.in
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India

Harsh Sharma*
harshsharma2024@gmail.com
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India

Vibhor Dave*
vibhor.dave03@gmail.com
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India

Abstract

With the increasing sophistication of password-guessing attacks, traditional user-generated passwords have become more susceptible to compromise, particularly due to predictable patterns and common substitutions that attackers exploit. This study explores the use of Large Language Models (LLMs) to generate robust passwords that are resistant to targeted guessing attacks. By comparing LLM-generated passwords with user-created passwords, this research aims to determine if LLMs can produce secure, memorable passwords that address common vulnerabilities. Using an online survey, we will collect both user-generated and LLM-generated passwords from participants and analyze their resistance to advanced guessing algorithms. The study's findings seek to contribute to cybersecurity practices by demonstrating whether LLMs can provide a practical solution for enhancing password security, offering insights that may guide future developments in secure password creation methods.

ACM Reference Format:
Manaswi Raj, Harsh Sharma, and Vibhor Dave. 2018. Leveraging LLMs to protect users against target password guessing. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (USP '24).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

In today's digital age, password-based authentication is still the primary security mechanism for most online platforms and services. However, user-generated passwords are often predictable and vulnerable to sophisticated password-guessing attacks. Attackers increasingly leverage machine learning (ML) and other data-driven methods to predict passwords based on common user tendencies, such as using easily memorable patterns or predictable substitutions (e.g., replacing "a" with "@"). Traditional password creation techniques fail to provide adequate protection against these advanced, targeted guessing attacks. This growing risk has highlighted the need for stronger, more unpredictable passwords that balance security with ease of recall.

Recent research has explored using Large Language Models (LLMs) for generating robust passwords that are resistant to such attacks [1]. Additionally, targeted online password guessing remains a significant security concern, as demonstrated by studies that examine the effectiveness of such attacks [2].

2 Research Questions

Our research questions were developed to address critical gaps in the understanding of LLM-generated passwords' effectiveness. They focus on two main aspects: the inherent guessability of LLM-generated passwords, and the influence of users' historical password data on the guessability of both LLM-generated and user-generated passwords.

Research Question 1

Does the LLM's password generation model, when tuned to avoid common user patterns, produce passwords that are statistically less guessable than user-created passwords derived from similar patterns?

Context: This question examines whether LLM-generated passwords, specifically tuned to avoid common user patterns, are statistically less guessable than passwords created by users. By comparing these two types of passwords, this study seeks to determine if LLMs can effectively minimize guessability by circumventing the predictable patterns users often employ.

Variables:

- **Password Type:** Categorical variable indicating if a password is LLM-generated (with tuning to avoid common patterns) or user-generated. This variable helps

*All authors contributed equally to this research.

USP '24, 10-09-24, Kharagpur, IN

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

differentiate and compare the guessability of each password source.
- **Guessability (targuess):** A continuous variable measuring the number of attempts needed by a targeted guessing algorithm to crack the password, representing password resilience.
- **Pattern Avoidance:** Binary variable indicating whether the password avoids common patterns (e.g., sequences, common substitutions). This variable is essential for examining the role of pattern avoidance in LLM-generated passwords.

Research Question 2

Does the number of previous passwords (n) provided by a user influence the guessability of an LLM-generated password, compared to the user's own subsequent password?

Context: This question investigates whether the quantity of prior passwords known to a targeted guessing algorithm impacts the security of LLM-generated passwords relative to user-generated ones. It aims to determine if a greater number of previous passwords affects the guessability of these two types of passwords differently, providing insight into how password history might impact targeted guessing effectiveness.

Variables:

- **Number of Previous Passwords (n):** A numerical variable representing the count of a user's prior passwords provided to the guessing algorithm, affecting its ability to predict subsequent passwords.
- **Password Type:** Categorical variable (LLM-generated or user-generated) used to compare the resilience of each password type.
- **Guessability (targuess):** A continuous variable that measures the number of attempts required to crack the password using targeted guessing methods.
- **Subsequent User Password:** Categorical variable for the user's password following their prior passwords, used for comparison against LLM-generated passwords to assess relative guessability.

3 Data Collection and Dataset Description

Data Collection

The dataset used in this research consists of 4.2 billion context-based passwords, collected from publicly available resources and password leak repositories. These sources include open datasets, password leaks, and breach disclosures that were accessed through ethical and legal means. The collection process strictly adhered to privacy guidelines and ethical research standards, ensuring that the data was obtained in compliance with relevant regulations. Special attention was given to aggregate and anonymize the data wherever possible. Each password entry was evaluated for its contextual relationship with accompanying information, such as email IDs or usernames, indicative of user behavior.

Dataset Description

The dataset is organized to provide insights into the contextual associations between user identifiers (e.g., email IDs) and their corresponding passwords. Each entry in the dataset includes:

- **Email ID:** The primary user identifier, stored in plain format to enable contextual analysis. During preprocessing, duplicate and invalid email entries were removed.
- **Password:** The associated password string, representing user preferences or patterns based on their contextual environment.

graphicx

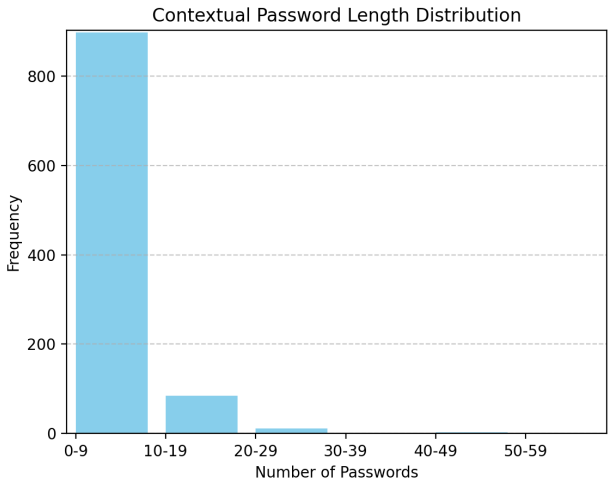

Figure 1. Distribution of Number of Passwords present

The dataset was thoroughly cleaned and structured into a relational format, facilitating the analysis of contextual trends, password strength evaluation, and potential vulnerabilities. Advanced techniques, such as tokenization and frequency analysis, were employed to identify common patterns and trends, providing insights into the predictability of passwords based on contextual information.

This dataset serves as a critical resource for understanding human password behavior and supporting the development of enhanced methods for password security and prediction.

4 Study Architecture

The study is designed to evaluate the effectiveness of LLM-generated passwords versus user-generated passwords by assessing their guessability, especially in cases where an attacker has prior knowledge of a user's password history. The

Table 1. Sample of Email and Password Dataset

Email ID	Password
user1@example.com	123456
john.doe@example.com	password123
alice.wonderland@email.com	alice2024
random.user@domain.com	qwerty123
test.email@provider.com	abcdef

study involves several stages, including data collection, password generation, and a comparative analysis using targeted guessing algorithms.

1. Data Collection

The dataset used in this research consists of 4.2 billion context-based passwords, collected from publicly available resources and password leak repositories.

2. Password Generation

Using the participant's personal information and previous password data, we will prompt an LLM to generate a password. The LLM will be fine-tuned to avoid common user patterns, leveraging the input data to produce a complex password that may or may not align with user tendencies.

3. Guessability Analysis using Targeted Guessing (targuess)

To assess the security of LLM generated passwords, we will use the ***targuess*** tool, which estimates the probability of successfully guessing a new password given previous password data.

4. Interpretation and Reporting

The results from the guessability analysis will be used to draw conclusions about the comparative effectiveness of LLM-generated and user-generated passwords. We will evaluate whether LLM-generated passwords offer significant security benefits and if they exhibit less predictability than user-created passwords based on similar data inputs. The findings will be compiled into a report that discusses the strengths and weaknesses of each approach and provides recommendations for enhancing password security using AI-generated methods.

5 System Architecture for Targeted Password Guessing Defense

To defend against targeted password guessing attacks, we have developed a system based on the large language model LLama3.2, which has 3 billion parameters. The architecture consists of two main components: the *Targuess* application and the *LLM Server* application, running on separate machines.

5.1 Client-Side: Targuess Integration and Dataset Parsing

On the client machine, a Python script performs the following tasks:

- **Parsing the Password Dataset:** The dataset consists of username-password pairs, in the format `<username>: <password1>, <password2>,<password3>`.
- **Interaction with Targuess:** The script sends the dataset entries to the *Targuess* application for simulating password-guessing attacks.
- **Request Forwarding to LLM Server:** Simultaneously, the dataset entries are forwarded to the LLM Server for password generation.

5.2 Server-Side: LLM Integration via Flask

The server-side consists of a Flask-based application that processes incoming requests:

- **Request Parsing:** The server receives the username-password pairs from the client and constructs a system prompt for the LLM.
- **LLama3.2 Execution:** The system prompt is passed to LLama3.2 via Ollama, which generates a more secure password.
- **Response Transmission:** The generated password is sent back to the client application.

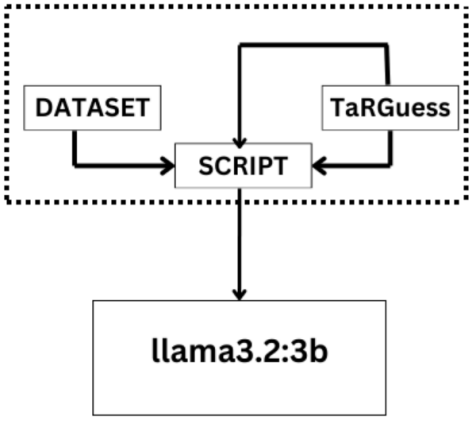

Figure 2. Simplified System Architecture

5.3 Client-Side Comparison and Accuracy Calculation

After receiving both the *Targuess* results and the LLM-generated password, the client compares them:

- **Comparing Outputs:** The client compares the guessability of the original passwords (from Targuess) with the new, more secure passwords generated by LLama3.2.

- **Accuracy Calculation:** The system tracks the accuracy over 10,00 data points, evaluating the effectiveness of the generated passwords.

5.4 Flexibility and Experimentation

Although the system is based on LLama3.2, it is designed to allow for easy substitution of other language models. This flexibility allows testing different models to find the most effective solution for password generation.

article amsmath

6 Result

Out of the 1000 passwords generated by LLama 3.2:3b, 187 passwords were successfully guessed by Targuess. This performance is summarized as follows:

$$\text{Passwords Generated} = 1000$$

$$\text{Passwords Guessed by Targuess} = 187$$

$$\text{Guessing Accuracy} = \frac{187}{1000} \times 100 = 18.7\%$$

Thus, Targuess was able to guess **18.7%** of the passwords generated by llama 3.2:3b. Although the result shows some success, the performance is not satisfactory. Achieving an accuracy of only 18.7% suggests that there is significant room for improvement in password geneartion phase. Ideally, we would aim for a lower success rate to demonstrate better predictive power and security assessment. Further optimization and fine-tuning of the password prediction model are necessary to decrease the accuracy of the attack.

graphicx

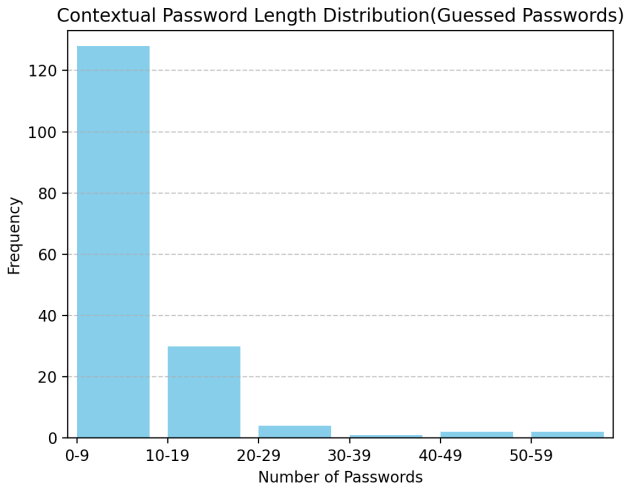

Figure 3. Distribution of Contextual Passwords present for the guessed passwords

7 Drawbacks of the Study

While the study provides valuable insights into the performance of the llama 3.3:3b model in password generation and the effectiveness of Targuess for password guessing, there are several drawbacks that need to be addressed:

- **Dependency on Leaked Datasets:** The study relies on publicly available datasets that may be compromised or incomplete. The quality of these datasets affects the overall results and might not represent real-world scenarios where passwords are more diverse or complex.
- **No Involvement of Real Users:** The passwords used in this study are generated and guessed based on predefined datasets. This lacks the involvement of real users, who would typically create passwords with a wider variety of patterns, making the results less applicable to actual user behavior.
- **Fine-Tuning Required:** The LLama 3.3:3b model and the Targuess algorithm require further fine-tuning to improve the accuracy of password generation and guessing. The current performance indicates that the models need better adaptation to the data and environment.
- **Trade-off Between Guessability and Memorability:** The generated passwords often strike a balance between guessability and memorability. While the generated passwords may be easier for users to remember, they could also be more predictable and easier to guess, compromising security. In our case the passwords were more memorable so they were comparatively Easy to Guess.

In conclusion, these drawbacks suggest that further work is needed to enhance the model's applicability to real-world scenarios, improve the security of generated passwords, and address the trade-offs involved.

References

[1] Rando, J., Perez-Cruz, F., and Hitaj, B. Passgpt: password modeling and (guided) generation with large language models. In *European Symposium on Research in Computer Security* (2023), Springer, pp. 164–183.

[2] Wang, D., Zhang, Z., Wang, P., Yan, J., and Huang, X. Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (2016), pp. 1242–1254.