Password Strength Checker

DISCRETE PROJECT

Today's Objectives

WHAT IS A PASSWORD?

The measure of how difficult it is for an attacker to guess or crack a password.

IMPORTANCE OF PASSWORD STRENGTH

Enhances security by increasing the difficulty of brute-force attacks.

OUR GOAL TODAY

Using discrete mathematics principles to analyze and implement a password strength checker.

Password Strength Checker

PROBABILITY

likelihood chancs of guessing

COMBINATIONS

How much are the total combinations in the password ?

PERMUTATION

Definition, concepts, relevant theories, and examples.

LOGICAL OPERATORS

Definition, concepts, relevant theories, and examples.

Combinations Of Passwords

- FOR A PASSWORD OF LENGHT N
  - LOWERCASE LETTERS: $(26)^N$
  - UPPERCASE LETTERS: $(26)^N$
  - DIGITS: $(10)^N$
  - SPECIAL CHARACTERS: E.G WE ARE GETTING 10 . SO $(10)^N$

- FOR EXAMPLE : IF WE ARE MAKING A PASSWORD OF 8 CHARACTERS THE TOTAL COMBINATIONS WILL BE

$$(26+26+10+10)^8 = 728$$

TOTAL POSSIBLE PASSWORDS: 722,204,136,308,736

Password Strength Criteria Using Logical Operators

AND OPERATOR (∧)

Password meets all requirements (e.g., uppercase and lowercase).

OR OPERATOR (∨)

Password has at least one requirement (e.g., either uppercase or digits).

NOT OPERATOR(!)

Exclude easy sequences (e.g., "123456").

PASSWORD IS STRONG IF: LENGTH ⩾ 8 AND CONTAINS UPPERCASE AND CONTAINS LOWERCASE AND CONTAINS SPECIAL CHARACTER

Probability of Guessing a Password

RANDOM GUESSING ATTACK

IF someone is guessing in the first try the probability of it will be

1/Total combinations

ENTROPY OF PASSWORD:

- Measures uncertainty:
  Entropy=log2(Total Combinations)
- A high entropy indicates a lower probability of guessing.

Complexity Analysis and Big-O Notation (worst case)

- FOR A PASSWORD OF LENGTH N, THE CHECKER RUNS IN O(N) AS IT EXAMINES EACH CHARACTER.

- WORST CASE: ALL CRITERIA CHECKED WITHOUT MEETING, STILL O(N).

Password Strength Checker Algorithm

CHECK LENGTH: ENSURE PASSWORD HAS A MINIMUM LENGTH N.

UPPERCASE CHECK: $\exists x \in \text{password} \mid x \in \text{uppercase}$

LOWERCASE CHECK: $\exists y \in \text{password} \mid y \in \text{lowercase}$

DIGITS CHECK: $\exists z \in \text{password} \mid z \in \text{digits}$

LOGICAL OUTPUT: PASSWORD PASSES IF IT MEETS ALL CRITERIA, INDICATING STRONG, MEDIUM, OR WEAK STATUS.

Real-World Applications and Future Considerations

- APPLICATIONS IN CYBERSECURITY
- ADAPTING TO MULTI-FACTOR AUTHENTICATION.

Conclusions

- Discrete mathematics provides tools for analyzing password strength.
- Password strength checking combines combinatorics, probability, and logical operations.
- Strong passwords can be systematically enforced by applying discrete mathematical principles.

Thanks!