Task 6: Create a Strong Password and Evaluate its Strength

Cybersecurity Lab Report

KANDI TEJASREE
ELEVATE LABS

Task 6:

Create a strong password and evaluate its strength

Index:

Objective:

Understand what makes a password strong by creating multiple passwords of varying complexity, testing them with online password strength checkers, and analyzing the results to identify best practices.

Tools Used:

- Passwordmeter.com
- Kaspersky password checker
- Norton password checker

Steps Performed:

- Firstly we write down atleast 4 -5 passwords.
- With length(short vs length) and character types(only lowercase, uppercase+lowercase, number, symbols, etc).

Example:

- password
- Password
- Password1
- p@s$word!
- $tr@ng^Pa$sW0rD%

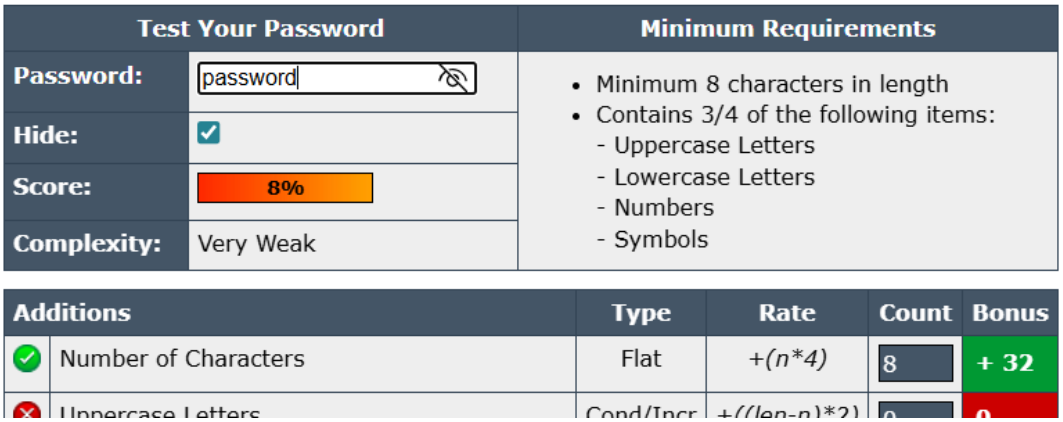
Now we have created some passwords for testing.

We shall choose a password strength checker tool for testing the passwords. We go for passwordmeter.com

Test Your Password		Minimum Requirements
Password:		• Minimum 8 characters in length • Contains 3/4 of the following items: - Uppercase Letters - Lowercase Letters - Numbers - Symbols
Hide:	☑	
Score:	0%	
Complexity:	Too Short	

Additions	Type	Rate	Count	Bonus
✗ Number of Characters	Flat	+(n*4)	0	0
✗ Uppercase Letters	Cond/Incr	+((len-n)*2)	0	0
✗ Lowercase Letters	Cond/Incr	+((len-n)*2)	0	0
✗ Numbers	Cond	+(n*4)	0	0
✗ Symbols	Flat	+(n*6)	0	0
✗ Middle Numbers or Symbols	Flat	+(n*2)	0	0
✗ Requirements	Flat	+(n*2)	0	0
Deductions				

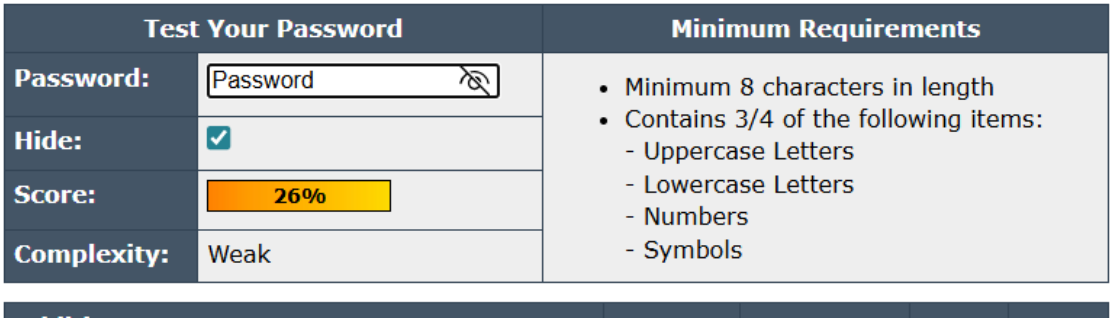
➢ Very weak: password



Feedback:-

- Password is very weak all are lowercase, common word, and very easy to guess or crack. So avoid of using dictionary words alone.

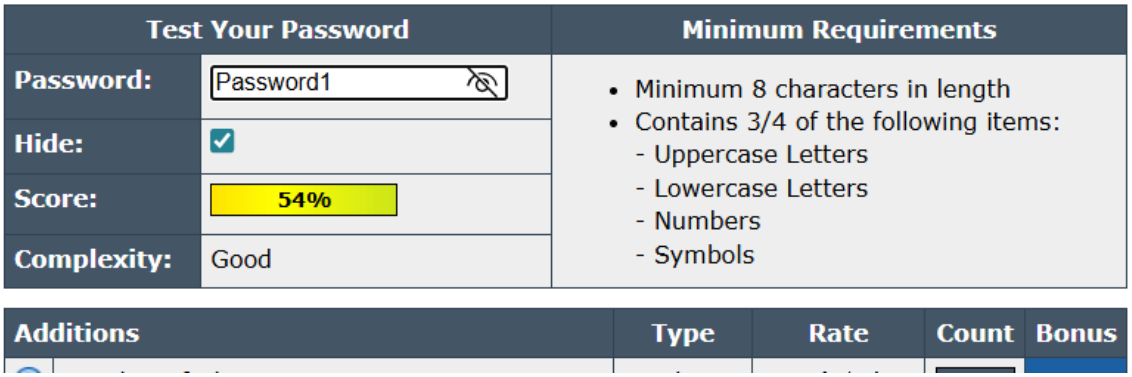
➢ Weak: Password



Feedback:-

- Slightly better due to uppercase of first letter but still very common and predictable. So adding of numbers or symbols and lengthening the password upto 8 characters.

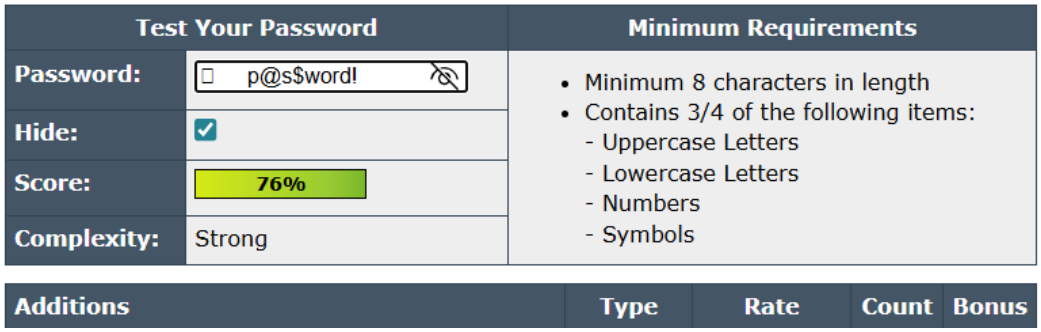
➢ Good: Password1



Feedback:-

- Better because it adds a number at the end but still mostly predictable and common patter. So need more symbols to add or length to be stronger

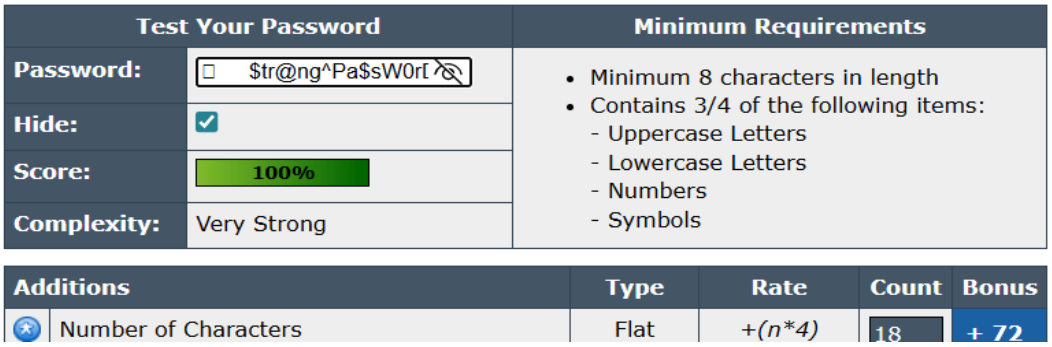
➢ Strong: p@s$word!



Feedback:-

- Much better because use of lowercase, symbols. Still based on a common word substituting words with more symbols.

➢ Very strong: $tr@ng^Pa$sW0rD%



Feedback:-

- Very Strong password mixes of uppercase, lowercase , number and multiple symbols making long and complex, much harder to crack.

Research of Common Password Attacks:

- **Brute Force Attack:** Tries every possible password combination until it finds the right one. Longer, complex passwords slow it down.

  Password Complexity:

  -Longer passwords increase the number of possible combinations exponentially.

  -Using uppercase, lowercase, numbers, and symbols increases the "keyspace" attackers must search.

  -This dramatically increases the time needed to crack the password.

- **Dictionary Attack:** Tries common words and passwords from a list. Simple or common passwords are easily guessed.

  Password Complexity:

  - Avoid simple dictionary words or common substitutions (password → p@ssword).
  - Use random or uncommon combinations of characters.
  - Longer, complex passwords are unlikely to be in any dictionary.

- **Credential Stuffing:** Uses leaked username/password pairs from one site to break into other sites where people reuse passwords.

  Password Complexity:

  -Using unique passwords for every account stops credential stuffing.

  -Complex, unique passwords are harder to reuse or guess.

- **Rainbow Table Attack:** Uses precomputed tables of password hashes to quickly reverse common passwords. Complex passwords are safer.

  Password Complexity:

  -Complex, long passwords generate hashes that are almost never in these tables.

  -Use of random characters greatly reduces vulnerability.

  -Salting passwords (adding random data before hashing) also prevents this, but that's usually done on the server side.

Conclusion:

Strong passwords with a mix of letters, numbers, and symbols protect against common attacks like brute force, dictionary, and credential stuffing. Using unique, complex passwords reduces the risk of being hacked. Testing passwords helps improve their strength and keep accounts safe.

We also researched common attack methods like brute-force and dictionary attacks, etc., highlighting how weak or reused passwords are highly vulnerable. Complexity increases a password's entropy, making it significantly harder to guess or crack.

In conclusion, creating strong, complex, and unique passwords is essential for protecting accounts and sensitive data. Implementing best practices—such as using 12+ character passwords with mixed character types, avoiding predictable patterns, and using password managers—can drastically improve cybersecurity. This task reinforces the importance of password hygiene as a foundational element of digital security.