

***Task 6 – Password Strength Evaluation***

***Author: Anshikaa Pahilajani***

***Cyber Security Internship — Elevate Labs***

***Date: 30 September 2025***

## Introduction

Passwords are the first line of defense for securing accounts and sensitive data. Weak passwords are highly vulnerable to attacks such as **brute force** and **dictionary attacks**, while strong passwords reduce these risks.

In this task, I created different types of passwords, tested their strength using the [tryzxcvbn demo \(Dropbox's zxcvbn estimator\)](#), and recorded the results.

## Password Testing Results

### Password 1: qwerty

#### demo

```
qwerty
password:      qwerty
guesses_log10: 0.69897
score:         0 / 4
function runtime (ms): 1
guess times:
100 / hour:    3 minutes      (throttled online attack)
10 / second:   less than a second (unthrottled online attack)
10k / second:  less than a second (offline attack, slow hash, many cores)
10B / second:  less than a second (offline attack, fast hash, many cores)
warning:       This is a top-10 common password
suggestions:   - Add another word or two. Uncommon words are better.
match sequence:
'qwerty'
pattern:       dictionary
guesses_log10: 0.60206
dictionary_name: passwords
rank:          4
reversed:      false
base-guesses:  4
uppercase-variations: 1
l33t-variations: 1
```

### Password 2: iloveyou

#### demo

```
iloveyou
password:      iloveyou
guesses_log10: 1.68124
score:         0 / 4
function runtime (ms): 1
guess times:
100 / hour:    29 minutes      (throttled online attack)
10 / second:   5 seconds       (unthrottled online attack)
10k / second:  less than a second (offline attack, slow hash, many cores)
10B / second:  less than a second (offline attack, fast hash, many cores)
warning:       This is a top-100 common password
suggestions:   - Add another word or two. Uncommon words are better.
match sequence:
'iloveyou'
pattern:       dictionary
guesses_log10: 1.6721
dictionary_name: passwords
rank:          47
reversed:      false
base-guesses:  47
uppercase-variations: 1
l33t-variations: 1
```

## Password 3: Admin@123

### demo

```
Admin@123
password:      Admin@123
guesses_log10: 7.45652
score:         2 / 4
function runtime (ms): 1
guess times:
100 / hour:    32 years      (throttled online attack)
10 / second:   1 month       (unthrottled online attack)
10k / second:  48 minutes    (offline attack, slow hash, many cores)
10B / second:  less than a second (offline attack, fast hash, many cores)
warning:       This is similar to a commonly used password
suggestions:   - Add another word or two. Uncommon words are better.
                - Capitalization doesn't help very much

match sequence:
'Admin'
pattern:       dictionary
guesses_log10: 3.15534
dictionary_name: passwords
rank:          715
reversed:      false
base-guesses:  715
uppercase-variations: 2
l33t-variations: 1
'@123'
pattern:       bruteforce
guesses_log10: 4
```

## Password 4: MSRUAS2025

### demo

```
MSRUAS2025
password:      MSRUAS2025
guesses_log10: 9.63414
score:         3 / 4
function runtime (ms): 1
guess times:
100 / hour:    centuries    (throttled online attack)
10 / second:   13 years     (unthrottled online attack)
10k / second:  5 days       (offline attack, slow hash, many cores)
10B / second:  less than a second (offline attack, fast hash, many cores)

match sequence:
'MSRUAS'
pattern:       bruteforce
guesses_log10: 6
'2025'
pattern:       spatial
guesses_log10: 3.33311
graph:         keypad
turns:         2
shifted count: 0
```

## Password 5: India#75

### demo

```
India#75
password:          India#75
guesses_log10:     6.11461
score:            2 / 4
function runtime (ms): 1
guess times:
100 / hour:       1 year          (throttled online attack)
10 / second:      2 days          (unthrottled online attack)
10k / second:     2 minutes       (offline attack, slow hash, many cores)
10B / second:     less than a second (offline attack, fast hash, many cores)
suggestions:
- Add another word or two. Uncommon words are better.
- Capitalization doesn't help very much

match sequence:
'India'
pattern:          dictionary
guesses_log10:    2.81023
dictionary_name:  english_wikipedia
rank:             323
reversed:         false
base-guesses:     323
uppercase-variations: 2
l33t-variations:  1
'#75'
pattern:          bruteforce
guesses_log10:    3
```

## Password 6: S@f3H0me!

### demo

```
S@f3H0me!
password:          S@f3H0me!
guesses_log10:     8.34971
score:            3 / 4
function runtime (ms): 0
guess times:
100 / hour:       centuries       (throttled online attack)
10 / second:      8 months        (unthrottled online attack)
10k / second:     6 hours         (offline attack, slow hash, many cores)
10B / second:     less than a second (offline attack, fast hash, many cores)

match sequence:
'S@f3'
pattern:          dictionary
guesses_log10:    3.50947
dictionary_name:  us_tv_and_film
rank:             404
reversed:         false
l33t subs:        3 -> e, @ -> a
un-l33ted:        safe
base-guesses:     404
uppercase-variations: 2
l33t-variations:  4
'H0me'
pattern:          dictionary
guesses_log10:    2.76343
dictionary_name:  english_wikipedia
rank:             145
reversed:         false
l33t subs:        0 -> o
un-l33ted:        home
base-guesses:     145
uppercase-variations: 2
l33t-variations:  2
'!'
pattern:          bruteforce
guesses_log10:    1.04139
```

Password 7: Tr@vel#Fun2025

demo

Tr@vel#Fun2025

```
password: Tr@vel#Fun2025
guesses_log10: 11.6925
score: 4 / 4
function runtime (ms): 2
guess times:
100 / hour: centuries (throttled online attack)
10 / second: centuries (unthrottled online attack)
10k / second: 2 years (offline attack, slow hash, many cores)
10B / second: 49 seconds (offline attack, fast hash, many cores)
match sequence:
'Tr@vel'
pattern: dictionary
guesses_log10: 3.58115
dictionary_name: passwords
rank: 953
reversed: false
l33t subs: @ -> a
un-l33ted: travel
base-guesses: 953
uppercase-variations: 2
l33t-variations: 2
'#Fun'
pattern: bruteforce
guesses_log10: 4
'2025'
pattern: spatial
guesses_log10: 3.33311
graph: keypad
turns: 2
shifted count: 0
```

Password 8: R3d!Lotus\$88

demo

R3d!Lotus\$88

```
password: R3d!Lotus$88
guesses_log10: 11.57664
score: 4 / 4
function runtime (ms): 1
guess times:
100 / hour: centuries (throttled online attack)
10 / second: centuries (unthrottled online attack)
10k / second: 1 year (offline attack, slow hash, many cores)
10B / second: 38 seconds (offline attack, fast hash, many cores)
match sequence:
'R3d!'
pattern: bruteforce
guesses_log10: 4
'Lotus'
pattern: dictionary
guesses_log10: 3.79837
dictionary_name: passwords
rank: 3143
reversed: false
base-guesses: 3143
uppercase-variations: 2
l33t-variations: 1
'$88'
pattern: bruteforce
guesses_log10: 3
```

Password 9: W!nter#Skies2025\*

demo

W!nter#Skies2025\*

```
password:           W!nter#Skies2025*
guesses_log10:      14.1726
score:              4 / 4
function runtime (ms): 2
guess times:
100 / hour:         centuries (throttled online attack)
10 / second:        centuries (unthrottled online attack)
10k / second:       centuries (offline attack, slow hash, many cores)
10B / second:       4 hours (offline attack, fast hash, many cores)
match sequence:
'W!nter'
pattern:            dictionary
guesses_log10:      2.87157
dictionary_name:    passwords
rank:               186
reversed:           false
l33t subs:          ! -> i
un-l33ted:          winter
base-guesses:       186
uppercase-variations: 2
l33t-variations:    2
'#Skies2025*'
pattern:            bruteforce
guesses_log10:      11
```

Password 10: MyC@t\$Sleeps!UnderTheSun#2025

MyC@t\$Sleeps!UnderTheSun#2025

```
password:           MyC@t$Sleeps!UnderTheSun#2025
guesses_log10:      25.45473
score:              4 / 4
function runtime (ms): 9
guess times:
100 / hour:         centuries (throttled online attack)
10 / second:        centuries (unthrottled online attack)
10k / second:       centuries (offline attack, slow hash, many cores)
10B / second:       centuries (offline attack, fast hash, many cores)
match sequence:
'MyC@t$'
pattern:            dictionary
guesses_log10:      5.94332
dictionary_name:    passwords
rank:               21941
reversed:           false
l33t subs:          @ -> a, $ -> s
un-l33ted:          mycats
base-guesses:       21941
uppercase-variations: 10
l33t-variations:    4
'Sleeps'
pattern:            dictionary
guesses_log10:      3.78661
dictionary_name:    us_tv_and_film
rank:               3059
reversed:           false
base-guesses:       3059
uppercase-variations: 2
l33t-variations:    1
'!'
pattern:            bruteforce
guesses_log10:      1.04139
'Under'
pattern:            dictionary
guesses_log10:      2.1271
dictionary_name:    english_wikipedia
rank:               67
reversed:           false
base-guesses:       67
uppercase-variations: 2
l33t-variations:    1
'The'
pattern:            dictionary
guesses_log10:      1.69897
dictionary_name:    english_wikipedia
rank:               1
reversed:           false
base-guesses:       1
uppercase-variations: 2
l33t-variations:    1
'Sun#2025'
pattern:            bruteforce
guesses_log10:      8
```

## Password Guidelines Learned

From the password strength tests, the following **best practices** were identified:

1. **Length matters** – passwords with 12+ characters are significantly stronger.
2. **Use all character types** – uppercase, lowercase, numbers, and symbols.
3. **Avoid dictionary words** – common words and phrases are easily guessed.
4. **Don't reuse passwords** across different accounts.
5. **Passphrases are powerful** – multiple random words with symbols/numbers are easy to remember but hard to crack.
6. **Update passwords regularly** to reduce risk from breaches.
7. **Use a password manager** to generate and store strong, unique passwords.
8. **Enable Multi-Factor Authentication (MFA)** for extra protection.

## Conclusion

Testing different passwords showed how quickly weak ones can fail against strength checks and how well-designed passwords resist attacks. By combining **length, randomness, and complexity**, and using **passphrases + MFA**, users can create secure passwords that protect accounts effectively.