Secure Password Analyzer

```
pip install requests
```

```
import hashlib
import requests
import re

common_passwords = [
    "123456", "password", "123456789", "12345678", "12345",
    "111111", "123123", "qwerty", "abc123", "password1"
]

# Function to check password strength
def check_password_strength(password):
    score = 0
    feedback = []

    # Length
    if len(password) >= 8:
        score += 1
    else:
        feedback.append("Password should be at least 8 characters long.")

    # Upper and lower case
    if re.search(r'[A-Z]', password) and re.search(r'[a-z]', password):
        score += 1
    else:
        feedback.append("Use a mix of uppercase and lowercase letters.")

    # Digits
    if re.search(r'[0-9]', password):
        score += 1
    else:
        feedback.append("Include at least one number.")

    # Special characters
    if re.search(r'[\W_]', password):
        score += 1
    else:
        feedback.append("Include at least one special character (e.g., !, @, #, etc.).")

    # Determine strength level
    if score == 4:
        level = "Strong Password"
    elif score >= 2:
        level = "Moderate Password"
    else:
        level = "Weak Password"

    numeric_score = score * 2.5  # Out of 10

    return level, feedback, numeric_score

# Function to check if password is found in data breaches
def check_pwned_password(password):
    sha1pwd = hashlib.sha1(password.encode('utf-8')).hexdigest().upper()
    prefix = sha1pwd[:5]
    suffix = sha1pwd[5:]

    url = f"https://api.pwnedpasswords.com/range/{prefix}"
    res = requests.get(url)

    if res.status_code != 200:
        return "Error checking breach status. Please try again later."

    hashes = res.text.splitlines()
    for line in hashes:
        hash_suffix, count = line.split(':')
        if hash_suffix == suffix:
            return f"This password has been found in {count} data breaches. Avoid using it."

    return "This password was not found in any known data breach."

# Main Program
if __name__ == "__main__":
    print("Password Strength and Breach Checker")
    print("Your password is never stored or transmitted beyond the secure API.\n")

    while True:
```

```
    password = input("Enter a password to check (or type 'exit' to quit): ")
    if password.lower() == 'exit':
        print("Exiting. Stay secure!")
        break

    # Check if password is common
    if password in common_passwords:
        print("Warning: This is a very common password. Avoid using it.")

    # Strength Check
    print("\n[1] Strength Analysis")
    strength, suggestions, score = check_password_strength(password)
    print(f"Result: {strength}")
    print(f"Strength Score: {score}/10")
    for tip in suggestions:
        print(f" - {tip}")

    # Breach Check
    print("\n[2] Breach Status")
    breach_result = check_pwned_password(password)
    print(f"Result: {breach_result}")

    print("\n" + "-" * 50 + "\n")
```

```
Password Strength and Breach Checker
Your password is never stored or transmitted beyond the secure API.

Enter a password to check (or type 'exit' to quit): hi@123

[1] Strength Analysis
Result: Moderate Password
Strength Score: 5.0/10
 - Password should be at least 8 characters long.
 - Use a mix of uppercase and lowercase letters.

[2] Breach Status
Result: This password has been found in 401 data breaches. Avoid using it.

--------------------------------------------------

Enter a password to check (or type 'exit' to quit): exit
Exiting. Stay secure!
```