Task 6: Create a Strong Password and Evaluate Its Strength.

1. Prepare workspace & install tools (Kali)

- \$mkdir -p ~/password-eval/{scripts,reports,screenshots}
- \$cd ~/password-eval
- \$sudo apt update

Install python3/pip if not present

\$sudo apt install -y python3 python3-pip openssl curl

Install zxcvbn python wrapper

- \$python3 -m pip install --user zxcvbn
- Or \$pip3 install zxcvbn-python

2. Create a set of test passwords (varied complexity)

- password123
- Summer2023!
- Tr0ub4dor&3
- correct horse battery staple
- G7!vR8#kPq2\$T9
- P@ssw0rd!P@ssw0rd!
- qwertyuiop
- L0ngButNo\$ymbols123
- Th!s is a long passphrase with spaces

Notes on types:

- **Very weak**: password123, qwertyuiop (common/dictionary)
- Medium: Summer2023! (mixed classes but predictable)
- Strong (random): G7!vR8#kPq2\$T9 (high entropy, random chars)
- **Passphrase**: correct horse battery staple or Th!s is a long passphrase... (long, memorable)

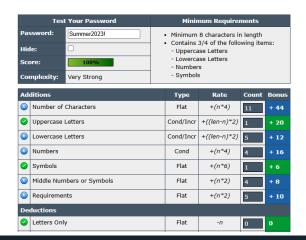
3. Evaluate passwords locally with zxcvbn (Python)

- Run Python Script and Check passwords
- If you use web checkers (**passwordmeter.com**, howsecureismypassword.net (https://www.security.org/how-secureis-my-password/)), do not enter real passwords.

The Password Meter Minimum 8 characters in length Contains 3/4 of the following items: Uppercase Letters Lowercase Letters password123 - Numbers - Symbols Number of Characters Flat +(n*4) 11 ♥ Uppercase Letters Cond/Incr +((len-n)*2) Lowercase Letters Cond/Incr +((len-n)*2) 3 Numbers Cond +(n*4) Symbols Flat +(n*6) 0 Middle Numbers or Symbols +(n*2)

The Password Meter

Letters Only



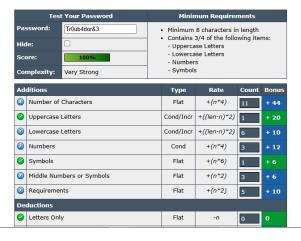
Flat

-n

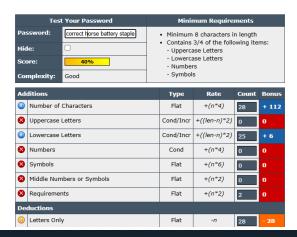
3

0

The Password Meter



The Password Meter



The Password Meter

