Password Strength Analyzer – Technical Brief

Overview The Password Strength Analyzer is a Python-based security analysis tool designed to demonstrate how modern systems evaluate password strength using entropy, pattern analysis, and heuristic rules. The project emphasizes education and explainability rather than authentication or storage.

No passwords are logged, stored, or transmitted.

Core Functionality

1. **Password Scoring Engine** Passwords are evaluated on a 0–4 scale, inspired by common strength meters in production systems. Scoring factors include:
   - Password length
   - Character class diversity (lowercase, uppercase, digits, symbols)
   - Repetition and sequence detection
   - Common password and dictionary pattern checks
2. **Entropy & Brute-Force Estimation** The analyzer estimates password entropy based on character set size and length, then translates entropy into a time-to-crack estimate under brute-force assumptions. Crack time model includes:
   - Offline attack scenarios
   - High-performance GPU-based guessing
   - Worst-case attacker assumptions Estimates are intentionally conservative to promote secure design thinking.
3. **Weakness Classification** Weak passwords are flagged with specific labels, including:
   - Insufficient length
   - Low entropy
   - Predictable structure
   - Common word or keyboard pattern usage Each weakness is mapped to remediation strategies.

4. **Passphrase Recommendation Engine** When weaknesses are detected, the system suggests high-entropy passphrases rather than complex but fragile passwords. Examples:
   - Multi-word constructions
   - Mixed case with numeric anchoring
   - Memorable but non-dictionary sequences This aligns with modern guidance favoring length over complexity.
5. **AI-Assisted Explainability Layer** An AI layer (Anthropic's Claude) generates factual, human-readable reports describing:
   - Current security status
   - Primary weaknesses
   - Recommended password patterns (2–3 examples, passphrase preferred) This layer focuses on education, not credential automation.

Security Design Principles

- No password persistence
- No hashing or authentication logic
- No network transmission beyond controlled API calls
- Input treated as ephemeral test data
- Designed strictly for offline learning and demos

Technologies & Concepts Demonstrated

- Python 3.11+ in Anaconda environment
- Entropy/pattern analysis (zxcvbn)
- Brute-force simulation models
- LLM integration (Anthropic API)
- Interactive UI (ipywidgets, Matplotlib visualization)
- Secure design: ephemeral data, dotenv secrets management

Learning Objectives This project demonstrates:

- Password entropy modeling
- Brute-force threat modeling
- Secure tool design without credential risk
- Translating technical security findings into human-readable explanations
- Balancing usability and security

Portfolio Context This analyzer is the first module in a broader Cybersecurity Portfolio, intended to showcase practical security concepts through interactive tools, simulations, and explainable AI components.

Author Robert Gravelle January 2026 Educational & portfolio use only