Brute Force vs Dictionary Attack

Brute Force: Systematically tries every possible character combination. It is slow but exhaustive.

Dictionary Attack: Leverages a list of pre-defined common passwords. It is extremely fast for weak passwords.

Why Weak Passwords Fail?

- **Dictionary Attacks:** Weak passwords (like "123456") are included in common wordlists like `rockyou.txt`. Attackers use tools like **John the Ripper** to compare your hash against these lists in seconds.
- **Low Complexity:** Short passwords have fewer possible combinations. This allows **Brute Force** attacks to try every possible variation quickly until the correct one is found.
- **Predictable Patterns:** Humans often use common substitutions (e.g., replacing 's' with '$') or personal info like birthdays, which are easily guessed or automated by modern cracking tools.
- **Fast Hashing:** Older algorithms like **MD5** are very fast. This is a weakness because it allows an attacker to test billions of password guesses per second against a stolen hash.
- **No "Salt":** Without a "salt" (random data added to the hash), identical passwords produce identical hashes. This makes it easy to crack many accounts at once using pre-computed tables.

Importance of MFA

Multi-Factor Authentication (MFA) is the most critical defense against password cracking. Even if an attacker successfully cracks a hash, the second factor (like a mobile app code) prevents unauthorized access.

Recommendations for strong authentication

Based on my analysis, I recommend the following for robust security:

- **Length:** Minimum of 12 characters to resist brute force.
- **Complexity:** Mix of uppercase, lowercase, numbers, and symbols.
- **Slow Hashing:** Use `bcrypt` instead of `MD5` or `SHA-1` to make cracking computationally expensive.