CS-029 · Passwords & Authentication

# Brute Force Attacks

Systematically trying many secrets until one works, defeated by length and rate limits.

## Executive Summary

A brute force attack systematically tries many possible secrets until the correct one is found. It can run online against a live login or offline against stolen password hashes. The defense is mathematical and procedural. Long, unpredictable secrets make the number of possibilities impractical, while rate limiting, lockout, slow hashing, and MFA make each attempt slow or unhelpful.

## What It Is

Brute force is the most direct guessing attack. In its purest form it tries every possible combination of characters until it lands on the right one, but pure exhaustion is rarely practical against long secrets, so attackers use smarter variants. A dictionary attack tries lists of likely passwords and common words. A hybrid attack adds predictable modifications such as appending numbers or swapping letters for symbols. Brute force comes in two settings. Online attacks submit guesses to a live system and are limited by its speed and defenses. Offline attacks run against stolen password hashes on the attacker's own hardware, where they can test enormous numbers of guesses per second.

## Why It Matters

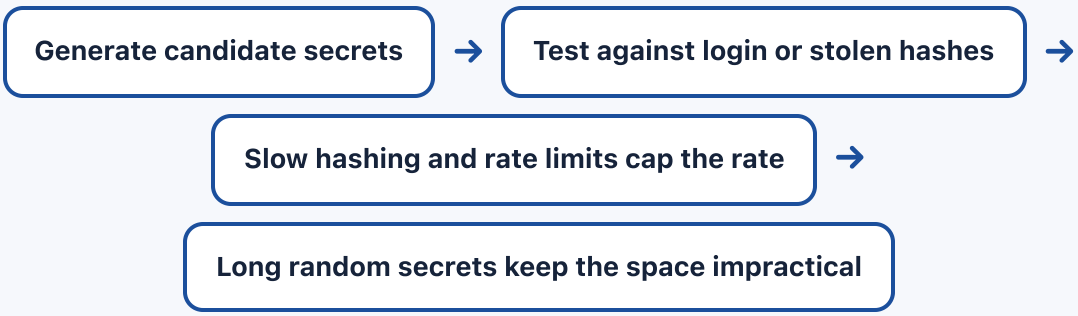
Brute force is a constant background threat and a benchmark for how strong a secret needs to be. Its feasibility depends entirely on the length and randomness of the secret and on how the system stores and rate limits attempts. Short or predictable passwords fall quickly, while long random ones remain out of reach even to fast offline cracking. Weak password storage, such as fast unsalted hashing, hands attackers a huge advantage after a breach. For a professional,

brute force explains why modern guidance stresses length, slow salted hashing, rate limiting, and MFA, and it is fundamental to reasoning about credential strength.

## How It Works

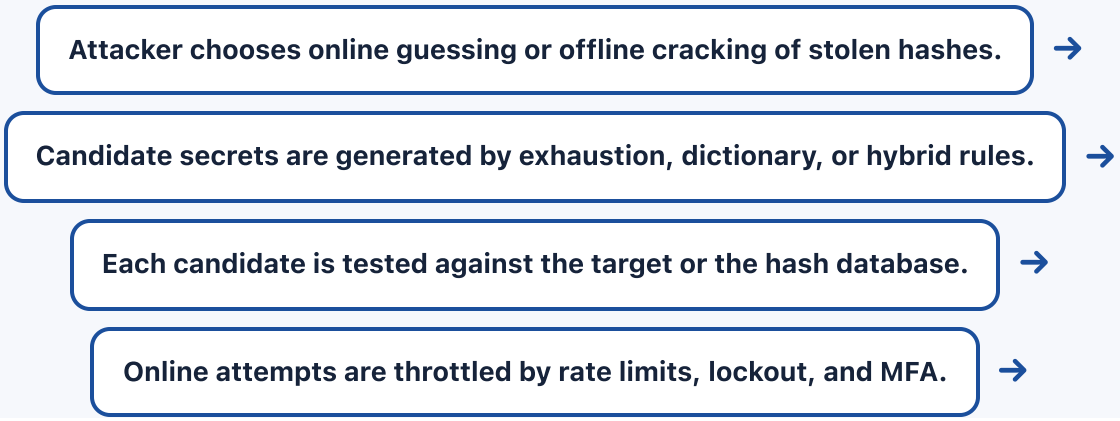
In an online attack, the attacker submits guesses to the login endpoint, so the system's rate limits, lockout, and MFA directly constrain how many guesses are possible and whether success even helps. This is slow and noisy, which is why attackers often prefer offline cracking. There they obtain a database of password hashes and run cracking software on powerful hardware, testing billions of candidates against the hashes. If the hashes were made with a fast function or without salts, cracking is far easier. Defenders raise the cost by using long secrets that expand the search space, slow memory hard salted hashing that limits guesses per second, and MFA so a cracked password alone is insufficient.

## Architecture Diagram




Guesses are generated and tested until one matches, so defenses aim to slow each attempt and enlarge the space.

## Visual Workflow



Offline attempts run at high speed unless slow salted hashing was used. →

Defenders enlarge the search space and slow each attempt to make it infeasible.

## Common Attacks

- Exhaustive guessing of short passwords and PINs
- Dictionary attacks using lists of common and breached passwords
- Hybrid attacks adding numbers, years, and symbol substitutions to words
- Offline cracking of stolen hashes on fast specialized hardware
- Brute forcing weakly protected remote access and administrative interfaces

## Common Mistakes

- Allowing short passwords that fall to exhaustion quickly
- Storing passwords with fast or unsalted hashing that accelerates cracking
- Having no rate limiting or lockout on login endpoints
- Exposing administrative and remote access interfaces to the open internet
- Relying on secrecy of the hash algorithm instead of slow salted hashing

## Best Practices

- Require long, unpredictable secrets to expand the search space
- Store passwords with a slow, salted, memory hard hashing function
- Apply rate limiting and lockout with backoff on login endpoints
- Require MFA so a cracked or guessed password is not sufficient
- Restrict and monitor remote access and administrative interfaces
- Screen against dictionaries and breached lists so easy guesses fail

## Quick Checklist

- [ ] Minimum length set high enough to resist exhaustion

- [x] Slow, salted password hashing confirmed in storage
- [x] Rate limiting and lockout with backoff on all logins
- [x] MFA enforced, especially on admin and remote access
- [x] Admin and remote interfaces restricted from the open internet
- [x] Common and breached password screening enabled

## Recommended Tools

**Rate limiting and lockout controls**
Slow or stop online guessing at the login endpoint

**Slow password hashing library**
Limits how many offline guesses per second are possible

**Breached and common password screening**
Blocks the dictionary candidates attackers try first

**Authenticator app or security key**
Ensures a cracked password alone cannot grant access

## Industry Standards

**NIST SP 800-63B**
Emphasizes length, breach screening, and rate limiting over forced complexity

**OWASP Authentication Cheat Sheet**
Guidance on password storage, rate limiting, and lockout

**OWASP Password Storage Cheat Sheet**
Specifies slow, salted, memory hard hashing to resist offline cracking

## Career Relevance

Brute force resistance is core to identity engineering, security operations, and penetration testing. Security engineers configure rate limiting, lockout, and slow hashing; SOC analysts detect and respond to guessing campaigns; penetration testers demonstrate weak credential

storage; and GRC analysts verify controls against frameworks. The mathematics of why length beats complexity, and why slow salted hashing matters, is foundational knowledge for the roles AI-Governance-Jobs.com serves.

## Interview Questions

- What is the difference between an online and an offline brute force attack?
- Why does password length matter more than adding a single special character?
- How does slow, salted hashing defend against offline cracking?
- Why is a dictionary attack usually more efficient than pure exhaustion?
- What controls limit online brute force, and how does MFA fit in?

## Related Certifications

CompTIA Security+ | ISC2 Certified in Cybersecurity (CC) | ISC2 SSCP

## Further Reading

- NIST SP 800-63B Digital Identity Guidelines
- OWASP Password Storage Cheat Sheet
- CISA: Use Strong Passwords

## Key Takeaways

- Brute force tries many secrets until one works, online or offline.
- Feasibility depends on secret length and how attempts are stored and limited.
- Long random secrets make exhaustion impractical.
- Slow, salted hashing sharply limits offline cracking speed.
- Rate limiting, lockout, and MFA blunt online guessing.

## FAQ

▶ **Why is offline brute force more dangerous than online?**

▶ **How long does a password need to be to resist brute force?**

▶ **Is a dictionary attack a kind of brute force?**

## Related Careers

### RELATED ROLES

Security Engineer | Soc Analyst | Iam Engineer

### RELATED CERTIFICATIONS

CompTIA Security+ | ISC2 Certified in Cybersecurity (CC) | ISC2 SSCP

### CURRENT OPENINGS

Live roles are on the job board.

**Browse all jobs**

### GUIDES & SALARY

Career guides | Role roadmaps | Salary data | Career tools

### SUGGESTED LEARNING PATH

1. Ground the basics with CS-001 Cybersecurity
2. Study this sheet: **Brute Force Attacks**
3. Go deeper: Passwords
4. Go deeper: Password Spraying
5. Validate it: work toward **CompTIA Security+**
6. Find the role: browse current openings

### RELATED SHEETS

CS-022 Passwords | CS-028 Password Spraying | CS-030 Rainbow Tables | CS-031 Account Lockout | CS-062 Authentication