B

# Brute Force Attack: Types, Warning Signs, and Prevention

A brute force attack repeatedly tests possible secrets until one works. Learn how online guessing differs from offline cracking, password spraying, and credential stuffing.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
5 min

CONTENTS

## In this guide

01 Types of password-guessing attacks
02 Online vs offline brute force
03 Warning signs
04 How to prevent brute force attacks
05 What to do when an attack is detected
06 How long does brute force take?
07 Frequently asked questions
08 Authoritative guidance

**A brute force attack repeatedly tests possible passwords, keys, PINs, tokens, or other secrets until a valid one is found.** Pure exhaustive search tries every possible combination. Real attackers usually improve efficiency with dictionaries, leaked-password patterns, rules, or previously stolen credentials.

Defenses depend on where guessing occurs. Online attacks interact with a login service and can be rate-limited and monitored. Offline attacks test guesses against stolen password hashes without contacting the service, so strong password hashing and long unique passwords become critical.

## 01 Types of password-guessing attacks

| Attack | How it works | Best defenses |
|---|---|---|
| Exhaustive brute force | Tries every possible value in a key space | Long secrets, rate limits, MFA, strong password hashing |
| Dictionary attack | Tests common words, passwords, and known patterns | Block weak and breached passwords; use password managers |
| Hybrid or rule-based attack | Mutates dictionary words with dates, substitutions, and suffixes | Long randomly generated passwords or passphrases |
| Password spraying | Tries a few common passwords across many accounts | Ban common passwords, MFA, cross-account detection |
| Credential stuffing | Tests username-password pairs stolen from another service | Unique passwords, breached-credential checks, MFA |
| Offline hash cracking | Tests guesses locally against stolen password hashes | Argon2id, scrypt, bcrypt, or PBKDF2 with salts and suitable cost |
| Cryptographic key search | Tests candidate encryption keys | Approved algorithms and sufficiently large keys |

Password spraying and credential stuffing are sometimes grouped with brute force because they automate repeated authentication attempts, but they do not enumerate every possible password. Distinguishing them improves detection and response.

## 02 Online vs offline brute force

An online attacker is constrained by network latency, throttling, lockouts, MFA, and monitoring. An offline attacker who has copied a password database can make large numbers of guesses privately. Salting prevents identical passwords from sharing the same stored value and defeats precomputed rainbow tables, while a memory-hard password hash makes every guess expensive.

Fast general-purpose hashes such as unsalted MD5 or SHA-1 are inappropriate for password storage. Encrypting a password database is not a substitute for password hashing because a stolen decryption key can expose every password directly.

## 03 Warning signs

- Many failed logins for one account, from one source, or across many accounts.
- One common password attempted against a large portion of the directory.
- Authentication attempts rotating through proxies, cloud hosts, devices, or geographic regions.
- Repeated MFA prompts, account lockouts, or successful login shortly after many failures.
- Large numbers of password-reset, recovery, API, SSH, RDP, VPN, or legacy-authentication requests.
- A new device or session followed by mailbox rules, token creation, downloads, or privilege changes.

Monitoring only one IP or one username misses distributed and low-and-slow attacks. Correlate account, source network, device, user agent, password pattern, protocol, and time.

## 04 How to prevent brute force attacks

1. **Use phishing-resistant MFA.** Passkeys and hardware-backed authenticators greatly reduce the value of a guessed password.
2. **Require long, unique passwords.** Let users paste from password managers and screen new passwords against known breached and common choices.
3. **Throttle intelligently.** Apply progressive delays and limits by account, source, device, network, and risk. A single fixed IP limit is easy to evade.
4. **Avoid permanent lockout as the only control.** Attackers can deliberately lock every account. Use temporary backoff, risk-based challenges, alerts, and secure recovery.
5. **Disable unnecessary authentication paths.** Remove legacy protocols, default accounts, exposed administration panels, and unused remote services.
6. **Store passwords safely.** Use a modern salted password-hashing function with a work factor calibrated for the service.
7. **Protect recovery.** Rate-limit resets, secure help-desk verification, rotate recovery codes, and do not let recovery bypass MFA.

## 05 What to do when an attack is detected

1. Preserve authentication, identity, application, WAF, VPN, and endpoint logs.
2. Identify targeted accounts, protocols, sources, time range, and any successful authentication.
3. Block or challenge confirmed malicious sources while enabling broader rate controls.
4. Revoke suspicious sessions and tokens and reset affected credentials through a trusted process.
5. Review successful accounts for mailbox rules, MFA changes, new keys, consent grants, downloads, privilege changes, and lateral movement.
6. Notify users and required security or privacy teams based on confirmed exposure.

Changing every user's password without establishing whether hashes or sessions were stolen can create disruption without closing the attacker path. Scope first, but contain confirmed access immediately.

## 06 How long does brute force take?

There is no universal answer. Time depends on the number of possible secrets, guessing speed, hashing cost, hardware, rate limits, and how predictable the secret is. A short password made from human patterns may fall early in a dictionary attack even if the theoretical character space appears large. Long randomly generated passwords and passphrases resist guessing far better.

## 07 Frequently asked questions

### Is credential stuffing a brute force attack?

It is an automated guessing attack using known username-password pairs rather than enumerating all combinations. Unique passwords stop reuse from succeeding.

### Does CAPTCHA stop brute force attacks?

It adds friction but is not sufficient alone. Attackers can distribute requests, use automation services, or target non-browser protocols. Combine it with MFA, rate limits, and monitoring.

### Can a strong password be brute-forced?

Any finite password can be guessed in theory. Sufficient length, randomness, secure hashing, throttling, and MFA can make the practical cost infeasible.

## 08 Authoritative guidance

- NIST: brute force password attack

- OWASP: Blocking Brute Force Attacks

NEED MORE HELP?

## We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

**Open this guide online at support.gridinsoft.com**