

Brute force protection



- [Introduction](#)
 - [How it works](#)
 - [Overview](#)
 - [Example: The login page](#)
 - [Usage](#)
 - [Activating](#)
 - [The brute force settings app](#)
 - [occ commands](#)
 - [Brute force protection and load balancers/reverse proxies](#)
 - [Troubleshooting](#)
 - [Overview](#)
 - [Excluding IP addresses from brute force protection](#)
 - [Brute force protection vs fail2ban](#)
-

Introduction

Nextcloud has built-in protection against brute force attempts.

The brute force protection feature is meant to protect Nextcloud servers from attempts to guess passwords and tokens in various ways. Besides the obvious "let's try a big list of commonly used passwords" attack, it also makes it harder to use slightly more sophisticated attacks via the reset password page or trying to find app password tokens. It is used throughout the Nextcloud ecosystem, including by other apps, if they have sensitive endpoints (and choose to enable support for it).

How it works

Overview

If triggered, brute force protection makes requests - coming from an IP address via a brute force protected endpoint - slower for up to a 24 hour period. In extreme circumstances it may prevent access outright, for up to 30 minutes, from a problematic IP address.

This protects your system from attackers trying, for example, a lot of different passwords.

The primary filter is IP address-based. This means that any account - even one associated with a given brute force attempt - is not impacted when it is connecting from a different IP address than any brute force attempts. This helps minimize inadvertent denial of service attacks against legitimate connections, while maximizing attack resistance from problematic IP sources.

Nuisance triggers are minimized through reasonable built-in defaults appropriate to each type of action.

The attempts history is automatically managed by a daily cronjob. Individual entries expire after 48 hours (attempts, however, may be still *logged* indefinitely elsewhere through the usual mechanisms within Nextcloud Server and at the discretion of the admin).

Excluding (whitelisting) select IP addresses from brute force protection to prevent false positives is supported, but usually false positives are best handled by fixing the underlying causes (e.g. a misconfigured reverse proxy or misbehaving client).



If you do notice a problem with the authentication behavior of any the official Nextcloud clients, please report it to the appropriate repository so that it can be looked into.

Keeping brute force protection active and operating properly helps protect your Nextcloud Server from malicious actors while minimizing potential impact on legitimate usage.

Example: The login page

The brute force protection is easiest to see in action on the login page. If you try to log in the first time with an invalid username and/or password you will not notice anything. But if you do this a few times you start to notice that the verification of the login is taking longer each time. This is the brute force protection kicking in.

The maximum delay is 25 seconds, unless maximum number of attempts (currently 10) was reached within the last 30 minutes (in which case a `429 Too Many Requests` will be returned until the

maximum attempts within the recent time has dropped below the threshold).

After a successful login (from the same source IP address), any prior invalid login attempts will be cleared and you will no longer be hit by the delay.

“ ”

Not all actions are necessarily viewed the same. It is possible for some activities to be more (or less) strict than others.

Usage

Activating

Brute force protection is enabled by default on Nextcloud. Its behavior can be adjusted through the `bruteforcesettings` app (shipped with Server and enabled by default), several `occ` commands, and several `config.php` parameters. Its effectiveness is highly dependent on having a properly configured environment, particularly when integrating a reverse proxy with Nextcloud (and associated parameters such as `trusted_proxies`).

The brute force settings app

This app, which shipped and enabled by default, makes it possible (via the Web UI) to view the status of a connection and modify certain parameters of the brute force protection built into Nextcloud Server.

The user interface added by this app is found under *Administration settings* -> *Security* under the *Brute-force IP whitelist* heading.

Currently an admin can view the status of the IP address they are connecting from as well as specify IPv4 or IPv6 addresses and ranges to exempt from brute force protection.

Additional enhancements may be made in the future, within this app and/or in combination with Nextcloud Server for additional monitoring or behavior adjustments related to brute force protection.

“ ”

Disabling the `bruteforcesettings` app does **not** disable brute force protection. The protection is built into Nextcloud Server core and is always active. Disabling the app only removes the ability to manage brute force settings from the Web interface.

“□

You would need to adjust the parameter `auth.bruteforce.protection.enabled` in your Nextcloud `config.php` to disable brute force protection, which is **heavily discouraged for production servers**, particularly if your server is reachable via a public IP address. It allows an attacker to iterate over all users and their passwords as well as two-factor verifications afterwards ultimately leading to admin access.

`occ` commands

There are several brute force related `occ` commands under `occ security`.

Brute force protection and load balancers/reverse proxies

If you are behind a reverse proxy or load balancer it is important you make sure it is setup properly. Especially the **trusted_proxies** and **forwarded_for_headers** `config.php` variables need to be set correctly. Otherwise it can happen that Nextcloud actually starts throttling all traffic coming from the reverse proxy or load balancer. For more information see [Reverse proxy](#).

Troubleshooting

Overview

On most setups Nextcloud will work out of the box without any issues. If you run into a situation where logging in or connecting is often very slow for multiple users, the first step is to check your Nextcloud Server logs to see what IP addresses are being detected (you will need to adjust your `loglevel` to `1` temporarily to do so).

Look for entries that start with any of the following:

- Bruteforce attempt from [...]
- IP address throttled [...]
- IP address blocked [...]

If all clients appear to be coming from the same IP address and that IP address happens to be your proxy, you need to review your `trusted_proxies` configuration.

If the IP address is a common connection point, such as a multi-user office location, it can be an option to whitelist it, with the draw back that users have to be trust-worthy.

For testing purposes you want want to whitelist your own IP address to see if the problem disappears. If it does - and assuming your proxy configuration is correct - you may have a client/device in your network that is misbehaving and generating invalid login attempts from your IP address.

You can use the `occ security:bruteforce:attempts` command to check the realtime status for a given IP address.

“ ”

The `bruteforce_attempts` database table will be empty if you're using a distributed memory cache since the database backend is no longer used unless it is the only option available.

Excluding IP addresses from brute force protection

“ ”

Most nuisance triggering of brute force protection can be resolved through proper configuration of reverse proxies. In other cases, select IP addresses that need to be whitelisted can be configured within this app (while leaving brute force protection enabled). This can be useful for testing purposes or when there are a lot of people (or devices) connecting from a known, single IP address.

It's possible to exclude IP addresses from the brute force protection.

- Make sure the `bruteforcesettings` app is enabled (it is by default)

- Login as admin and go to **Administration settings -> Security**

“ ”

Any excluded IP address can perform authentication attempts without any throttling. It's best to exclude as few IP addresses as you can, or even none at all.

Brute force protection vs fail2ban

Nextcloud's built-in brute force protection and fail2ban are complementary tools that operate at different layers of the stack. Using both together is recommended for production servers.

Nextcloud brute force protection is built into Nextcloud Server itself (the `bruteforcesettings` app provides the admin UI and exclusion settings, but the protection runs regardless). It works at the **application layer**: it detects suspicious login patterns and adds progressively longer delays to requests from the offending IP address. It has full context about Nextcloud-specific endpoints and credentials, and it activates automatically without any operating system configuration.

fail2ban works at the **OS/network layer**. It watches log files for failed login entries and instructs the system firewall (e.g. `iptables` or `nftables`) to block the offending IP outright. Blocked requests are dropped before they reach the web server, PHP, or the database, saving server resources entirely.

The two approaches are not mutually exclusive:

- Nextcloud brute force protection handles application-level throttling transparently, including for API clients and mobile apps, with no system configuration required.
- fail2ban reduces server load by blocking repeat offenders at the network level before their requests consume any application resources.

For setup instructions for fail2ban with Nextcloud, see [Setup fail2ban](#).

#3

cocobook-sync 2026-10-28 16:26:58 UTC

cocobook-sync 2026-10-29 01:09:16 UTC