POWER-USER

How To Run A Local Password Vault With KeePassXC

Plain-English how-to. ~60 minutes. Cloud-free password management for the privacy-focused.

Cloud password managers are great for most people. But if you've been following the recent vendor breaches (LastPass, Authy, others), or you simply prefer your most sensitive data not stored on someone else's servers, there's a better option: **KeePassXC**. Free, open-source, audited, fully local, and works on every OS.

The trade-off is that you manage your own sync, backups, and platform setup. The reward is a password vault that no third-party breach can ever expose, that you can audit line-by-line if you want, and that costs nothing.

By the end of this guide you'll have a KeePassXC vault running on your computer, browser integration set up, mobile access via Keepass2Android or KeePassium, and a sync strategy that keeps everything in sync without exposing the vault.

Quick snapshot

What you'll learn	Install KeePassXC, create a strong vault, enable browser + mobile access, and set up safe sync.
Skill level	Intermediate
Time required	60 minutes
What you'll need	A computer (Windows / Mac / Linux), a phone, a cloud sync provider (your choice)
Risk if you skip this	Reliance on a cloud vendor that could be breached or shut down
PDF kit	■ Download at the bottom of this page

Why this matters

Cloud password managers are convenient, but every one of them is a high-value target for attackers. LastPass's 2022 breach exposed encrypted vault data; users with weak master passwords had their vaults cracked offline.

A local vault stored in a file you control is fundamentally different: there's no remote server to breach, no API surface to attack, and the file itself is encrypted with a strong key derivation function (Argon2) that resists brute force even on stolen copies.

The cost is operational complexity — you have to handle sync and backups yourself. For the right user, the trade is worth it. KeePassXC has been independently audited and is the gold-standard for offline password

management.

Before you start

Decide on your sync strategy. Options: a cloud storage service (Dropbox, OneDrive, iCloud, Sync.com, Tresorit), a Syncthing-based mesh, or USB-only (most secure, least convenient). The vault file is encrypted regardless.

Pick a strong master password. 5-7 random words from a wordlist (correct-horse-battery-staple style) is far better than complex 12-character mush. Write it on paper and store it physically until memorized.

Have a way to safely back up the master password. KeePassXC supports a key file as an additional factor — losing both vault password and key file means permanent data loss.

Step 1 — Install KeePassXC

Download from **keepassxc.org** (Windows installer, macOS .dmg, Linux package). Verify the GPG signature on the download — KeePassXC publishes signing keys; check the download against them.

On Linux, install via your package manager: apt install keepassxc or brew install keepassxc on macOS.

Step 2 — Create your first database

File → New Database. Name it something benign ('personal.kdbx'). Set encryption: leave Argon2id (the secure default). Adjust transform rounds upward until your computer takes ~1 second — slows brute-force attacks proportionally.

Set the master password. KeePassXC's strength meter is honest — aim for 'Good' or better. Optionally add a key file (recommended).

Step 3 — Generate a key file (optional but recommended)

Tools → Generate Key File. Save to a USB stick or local-only folder. The key file is required in addition to your master password to open the vault.

Without the key file, even a leaked master password is insufficient. Store the key file separately from the vault file (e.g., USB stick + cloud sync of vault file).

Step 4 — Install the browser extension

KeePassXC-Browser is available for Chrome, Firefox, Edge, and Brave. After installing: in KeePassXC, **Tools** → **Settings** → **Browser Integration** → **Enable**. Choose your browsers.

Visit a site, sign in, KeePassXC will prompt to save. Auto-fill works similarly to commercial password managers.

Step 5 — Set up mobile access

iOS: **KeePassium** (free + paid tiers, FOSS). Android: **KeePass2Android** (free, FOSS). Both read .kdbx files.

Point the mobile app at your synced vault location (Dropbox / OneDrive / iCloud Files). The app downloads the encrypted file and unlocks it locally.

Step 6 — Sync the vault file safely

Easiest: drop the .kdbx file into Dropbox / OneDrive / iCloud Drive / Sync.com. The provider only ever sees encrypted bytes; cannot read your data.

More private: Syncthing — peer-to-peer file sync between your devices, no cloud at all. Requires running Syncthing on each device but works fantastically.

Step 7 — Migrate from your current password manager

Export from your current manager (1Password / Bitwarden / LastPass) as CSV. In KeePassXC: Database → Import → CSV. Map the columns, import.

Crucial: securely delete the CSV after import. CSVs contain plaintext passwords; never leave one on disk.

Step 8 — Set up automated backups

Vault file = critical data. Back up the .kdbx file daily to a location separate from your sync provider: external drive, secondary cloud (Backblaze B2), or a private NAS.

Test recovery: take a backup, copy to another machine, try to open with your master password + key file. Confirm it works before you depend on it.

PRO TIP

Master Password + Key File = Real Security.

Just a master password is fine but a key file adds a second factor that's hard to phish.

Store the key file separately from the vault file.

Argon2id key derivation slows offline brute-force attacks dramatically.

Backups separate from sync — restore-tested at least once a year.

If you want to go further: power-user upgrades

Power-user upgrade #1 — Use a YubiKey challenge-response slot

KeePassXC supports HMAC-SHA1 challenge-response with a YubiKey as an additional unlock factor.

Trade-off: requires the YubiKey to be present every time.

Power-user upgrade #2 — Run multiple databases for different domains

Personal.kdbx, Work.kdbx, Family.kdbx. Compromise of one doesn't expose others.

Trade-off: more state to manage.

Power-user upgrade #3 — Use Syncthing for true cloudless sync

Encrypted peer-to-peer sync with no third party involved.

Trade-off: must run Syncthing on every device.

Power-user upgrade #4 — Automate vault backups with a script

Cron job: nightly copy of .kdbx to local backup, weekly to remote cloud, monthly to offline drive.

Trade-off: script setup.

Power-user upgrade #5 — Use KeePassXC SSH agent

Store SSH private keys inside the encrypted vault; use KeePassXC as ssh-agent. Eliminates key files on disk.

Trade-off: vault must be open during SSH use.

Power-user upgrade #6 — Audit the vault with KeePassXC's health check

Database → Database Reports → Passwords. Lists reused, weak, and pwned passwords.

Trade-off: requires occasional cleanup time.

Common mistakes & pitfalls

Mistake — Weak master password.

Fix — 5+ random words minimum. No reused or guessable mush.

Mistake — No backups of the vault file.

Fix — Sync isn't backup. Always have independent restore copies.

Mistake — Storing key file in same place as vault.

Fix — Defeats the purpose of having one.

Mistake — Skipping audit of imported CSV.

Fix — Old reused passwords come along. Rotate during migration.

Mistake — Forgetting to remove the import CSV.

Fix — Plaintext passwords on disk are catastrophic.

Mistake — Trusting the browser plugin to be perfectly secure.

Fix — It's audited but a browser compromise can still read what's auto-filled in the page.

Mistake — Sharing the .kdbx file via email or chat.

Fix — Encrypted, but unnecessary attack surface. Use private sync.

Pro tips

Pro tip 1. Use the password generator to make all your replacements long, random, unique.

Pro tip 2. Tag entries by category — work, personal, finance — for quick filtering.

Pro tip 3. The TOTP fields in KeePassXC double as authenticator app for accounts that use TOTP.

Pro tip 4. Set per-entry expiry dates for accounts you should rotate annually.

Pro tip 5. Once a year, run the Database Reports → Passwords audit and clean up reused/weak entries.

Frequently asked questions

Is KeePassXC really safer than 1Password or Bitwarden?

Different threat models. Commercial cloud managers are easier and protect against your own backup failures. KeePassXC protects against vendor breaches and surveillance. Pick based on what you most fear.

Can my family share a KeePassXC vault?

Yes — keep a 'family' vault in a shared sync folder. Everyone with the master password + key file can open it.

What happens if I forget my master password?

Data is permanently lost. KeePassXC has no recovery. This is by design — also why you should consider a written/printed backup of the password stored physically.

Does KeePassXC work with face/Touch ID?

Mobile apps (KeePassium, KeePass2Android) support biometric unlock after the first password entry. Desktop KeePassXC supports Touch ID on macOS in recent versions.

What's the difference between .kdbx and .kdb?

.kdbx is the modern (KeePass 2) format with stronger encryption. Always use .kdbx.

Will my browser auto-fill still work?

Yes — KeePassXC-Browser integrates with all major browsers.

Is the source code really audited?

Yes — most recently in 2023 by Cure53. Audit reports are publicly available.

Quick recap — do these in order

DO THIS RIGHT NOW

The 8-step recap.

1. Install KeePassXC from keepassxc.org and verify the signature.
2. Create a .kdbx vault with a strong master password.
3. Generate and separately store a key file.
4. Install the browser extension + mobile app.
5. Sync via cloud storage or Syncthing.
6. Migrate from your current manager via CSV (then delete it).
7. Set up automated, independent backups.
8. Audit annually with Database Reports.

Mini glossary

.kdbx: Modern KeePass 2 database file format.

Argon2id: Memory-hard key derivation function used by KeePassXC.

Key file: An additional unlock factor stored separately from the vault.

CSV import: Bulk-load entries from another password manager.

Syncthing: Peer-to-peer file sync without a cloud provider.

Database Reports: KeePassXC's built-in vault health audit.