

AI + CREDENTIALS SAFETY CHECKLIST

How to let AI agents like Claude Code and Cowork touch real accounts and automate real work — without ever handing over the keys.

Companion to "I Let AI Into My Accounts. It Never Sees a Password." — chumania.com

THE ONE RULE

Never let an AI — or yourself, in a hurry — hand-type a password or paste a raw API key into a chat, script, or file. Every credential goes through the password manager. No exceptions, no "just this once."

PASSWORDS !' LOGIN ITEMS

Service name + username + a generated password you never actually see. Autofilled directly at the real login screen — never typed, never copied, never in a clipboard history.

API KEYS !' CREDENTIAL ITEMS

Paste the key in once, label it clearly, then reference it by its vault path from scripts and agents. It never gets pasted into the code itself.

```
op://vault/service-name/credential
```

AUTOMATION GETS ITS OWN ACCOUNT

A separate account with scoped, limited access — never your personal login. If one token is ever compromised, revoke it alone. Your real accounts were never in the blast radius.

LABEL EVERYTHING

Convention: [Service] – [What it's for] – [Date created]. An unlabeled key is a liability even sitting safely in a vault — you can't tell what breaks if you kill it.

```
Stripe API - checkout automation - Jan 2026
```

ROTATE ON A SCHEDULE

High-value keys (payments, admin access) rotate on a fixed calendar — no incident required. Any suspected exposure means rotate immediately. "Probably fine" is not a rotation policy.

APPROVE EVERY REQUEST INDIVIDUALLY

Don't rubber-stamp credential requests. The safety of the whole system depends on actually looking at what's being asked for, every time.

NEVER LET A COPY ESCAPE THE VAULT

No screenshots of keys, no passwords pasted into notes "just for now." The vault only protects what actually stays inside it.