

How to Setup a...

- [Using KeePassXC as the Secret Service](#)
- [How to Setup a YubiKey Authentication](#)

Using KeePassXC as the Secret Service

[KeePassXC_Project keepassxreboot/keepassxc](#)

Installation

```
sudo apt install keepassxc
```

Create a D-Bus Service That Implements the Secret Service API

```
mkdir -p "$HOME/.local/share/dbus-1/services"
```

```
tee "$HOME/.local/share/dbus-1/services/org.freedesktop.secrets.service"
>/dev/null <<'EOF'
[D-BUS Service]
Name=org.freedesktop.secrets
Exec=/usr/bin/keepassxc
EOF
```

Disable the GNOME Keyring Daemon Service

```
systemctl --user mask gnome-keyring-daemon.socket
systemctl --user mask gnome-keyring-daemon.service
sudo systemctl mask gnome-keyring-daemon.socket
sudo systemctl mask gnome-keyring-daemon.service
```

```
mkdir -p "$HOME/.config/autostart/"
cp "/etc/xdg/autostart/gnome-keyring-pkcs11.desktop" "$HOME/.config/autostart/"
cp "/etc/xdg/autostart/gnome-keyring-secrets.desktop"
"$HOME/.config/autostart/"
cp "/etc/xdg/autostart/gnome-keyring-ssh.desktop" "$HOME/.config/autostart/"
echo "Hidden=true" >> "$HOME/.config/autostart/gnome-keyring-pkcs11.desktop"
echo "Hidden=true" >> "$HOME/.config/autostart/gnome-keyring-secrets.desktop"
echo "Hidden=true" >> "$HOME/.config/autostart/gnome-keyring-ssh.desktop"
```

Until an appropriate solution is found, resort to removing "executable" permission from `gnome-keyring-daemon`.

```
sudo chmod -x /usr/bin/gnome-keyring-daemon
```

```
sudo tee "/etc/apt/apt.conf.d/99-disable-gnome-keyring-daemon" >/dev/null <<'EOF'
DPkg::Post-Invoke {
    "if [ -e /usr/bin/gnome-keyring-daemon ]; then chmod -x /usr/bin/gnome-
keyring-daemon; fi";
};
EOF
```

Unset GNOME Keyring as the Chromium Password Store

```
sudo tee --append "/etc/chromium.d/default-flags" >/dev/null <<'EOF'

# Disable GNOME keyring as password store (using keepassxc)
export CHROMIUM_FLAGS="$CHROMIUM_FLAGS --password-store=basic"
EOF
```

Configuration

- General
 - Basic Settings
 - Startup
 - ☒ Start only a single instance of KeePassXC
 - ☒ Automatically launch KeePassXC at system startup
 - ☒ Minimize window at application startup
 - ☐ Minimize window after unlocking database
 - ☒ Remember previously used databases
 - ☐ Load previously open databases on startup
 - ☒ Remember database key files and security dongles

Unlocking the Keyring Automatically

Create a KeePassXC Database Protected by Keyfile

```
mkdir -p --mode=700 "$HOME/.secrets/databases"
mkdir -p --mode=700 "$HOME/.secrets/private"
keepassxc-cli db-create --decryption-time 1000 --set-key-file
"$HOME/.secrets/private/SecretService.keyx"
"$HOME/.secrets/databases/SecretService.kdbx"
```

Encrypt the Keyfile Using the TPM2-Sealed Key

```
sudo systemd-creds encrypt --name=KeePassXC-SecretService --with-key=tpm2
"$HOME/.secrets/SecretService.keyx" "$HOME/.secrets/SecretService.creds"
sudo chown $USER:$USER "$HOME/.secrets/SecretService.creds"
sudo chmod 0400 "$HOME/.secrets/SecretService.creds"
```

⚠ Ensure you have a backup copy of the keyfile stored securely before shredding it — there is no recovery option if the TPM-sealed key becomes unusable (e.g. due to a firmware update).

```
shred "$HOME/.secrets/SecretService.keyx"
```

Authorize Credentials Decryption Using the TPM2-Sealed Key

```
sudo tee "/etc/polkit-1/rules.d/49-systemd-creds.rules" >/dev/null <<'EOF'
polkit.addRule(function(action, subject) {
    if (action.id == "io.systemd.credentials.decrypt" &&
        subject.local == true && subject.active == true &&
        subject.isInGroup ("tss")) {
        return polkit.Result.YES;
    }
});
EOF
```

```
sudo gpasswd --add <username> tss
```

Set Up Automatic Unlock on Login

```
sudo wget --quiet -O "/usr/local/libexec/keepassxc-watch.sh"  
"https://kb.havlas.me/attachments/16"  
sudo chown root:root "/usr/local/libexec/keepassxc-watch.sh"  
sudo chmod 0755 "/usr/local/libexec/keepassxc-watch.sh"
```

```
sudo tee "/etc/systemd/user/keepassxc-watch@.service" >/dev/null <<'EOF'  
[Unit]  
Description=Auto-unlock KeePassXC database '%i' via TPM2-sealed key  
After=graphical-session.target  
ConditionPathExists=%h/.secrets/%i.kdbx  
ConditionPathExists=%h/.secrets/%i.creds  
  
[Service]  
ExecStart=/usr/local/libexec/keepassxc-watch.sh %i  
Restart=on-failure  
RestartSec=2  
  
[Install]  
WantedBy=graphical-session.target  
EOF
```

```
sudo chown root:root "/etc/systemd/user/keepassxc-watch@.service"  
sudo chmod 0644 "/etc/systemd/user/keepassxc-watch@.service"  
sudo systemctl daemon-reload
```

```
systemctl --user enable --now keepassxc-watch@SecretService.service
```

SSH Keys

```
sudo apt install seahorse  
sudo mkdir -p /usr/libexec/ssh  
sudo ln -s ../seahorse/ssh-askpass /usr/libexec/ssh/ssh-askpass
```

```
[Desktop Entry]  
Type=Application  
Name=SSH ask-pass  
Exec=/usr/libexec/ssh/ssh-askpass  
StartupWMClass=ssh-askpass  
NoDisplay=true
```

```
systemctl edit --user ssh-agent.service
```

```
[Service]
```

```
#Environment=DISPLAY=:0
```

```
Environment=SSH_ASKPASS=/usr/libexec/ssh/ssh-askpass
```

```
systemctl --user daemon-reload
```

```
systemctl --user restart ssh-agent.service
```

How to Setup a YubiKey Authentication

[Yubico](#) [Yubico documentation](#) [Buy at Alza.sk](#)

Installation

```
sudo apt install fido2-tools libpam-u2f pcscd
```

```
sudo apt install yubikey-manager yubikey-personalization
```

Configuration

Set YubiKey PIN

```
ykman fido access change-pin
```

Disable YubiKey OTP Protocol

```
ykman config usb --disable OTP
```

Setup GNOME/PAM Authentication

```
auth          sufficient  pam_u2f.so cue [cue_prompt=? Tap the security key  
now...] openasuser userpresence=1
```

```
@include common-yubico
```

```
@include common-yubico
```

```
auth            sufficient    pam_u2f.so nodetect cue [cue_prompt=? Tap the security  
key now...] openasuser
```

```
mkdir -p ~/.config/Yubico  
pamu2fcfg > ~/.config/Yubico/u2f_keys
```

Setup Full Disk Encryption

```
sudo apt install dracut dracut-core
```

```
sudo tee "/etc/dracut.conf.d/60hostonly.conf" >/dev/null <<'EOF'  
hostonly=yes  
hostonly_cmdline=yes  
EOF
```

```
sudo dracut --regenerate-all --force
```

```
sudo systemd-cryptenroll [PARTITION] --fido2-device=auto --fido2-with-client-  
pin=no
```