
krapplet

Release 0.4.0

Johannes Willem (Hans) Fernhout

Jun 11, 2023

CONTENTS

1	Table of contents	3
1.1	Introduction	3
1.2	Screenshots	4
1.3	Installation	8
1.4	Usage	9
1.5	Storage providers	10
1.6	Disclaimers	11
1.7	Troubleshooting	12
1.8	License	12
1.9	Revision history	13

Krapplet is a graphical password manager for Linux. Its name derives from 'keyring applet'. Originally krapplet was purely built as a frontend for gnome-keyring, but now it also supports another mechanism for storing passwords, a format compatible with the [pass](#) command line password manager.

TABLE OF CONTENTS

1.1 Introduction

1.1.1 Why use a password manager

In today's on-line world, people need to maintain lots of passwords, particularly to access websites. These passwords need to have a certain complexity, so that it is hard to guess them, or crack them. However, complex passwords are hard to remember, and therefore people write them down on a piece of paper or in text files. Such written down passwords are not very secure though. Password managers were invented to store complex passwords in a safe and secure manner.

1.1.2 Krapplet goals and features

The aim of krapplet is to be a Linux native password manager. Krapplet provides the following features:

- simple to use
- sitting in the systray: ready to use whenever you need it
- tries not to be in the user's way; uses only a small amount of user's computer screen
- flexible: store also associated information, like a username, e-mail address, and the website URL
- uses common Linux manners to store and secure passwords, either gnome-keyring or GPG
- built in password generator

1.1.3 Concept

Password aka secrets need to be stored in a safe manner, i.e. encrypted. A well established mechanism for that in the Linux world is [gnome-keyring](#). Gnome-keyring organizes secrets in keyrings, often two keyrings exist:

- Login: which can be opened once a user logs in, and in which a user can store the secrets to open other keyrings.
- Default: typically used for application secrets

Krapplet builds on this concept, allowing not only passwords to be stored, but also other information like username, url, email address, et

Alternatively, krapplet also support the storage format adopted by [pass](#), which encrypts key files in a directory structure under `${HOME}/.password-store` using [GunPG](#).

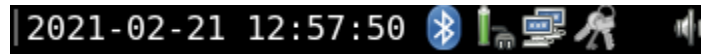
1.1.4 Environment

Krapplet is currently a Linux only application. Krapplet has been tested on the amd64 and x64 architecture, but might work on other architectures as well. A system tray needs to be available for the applet to embed itself in.

1.2 Screenshots

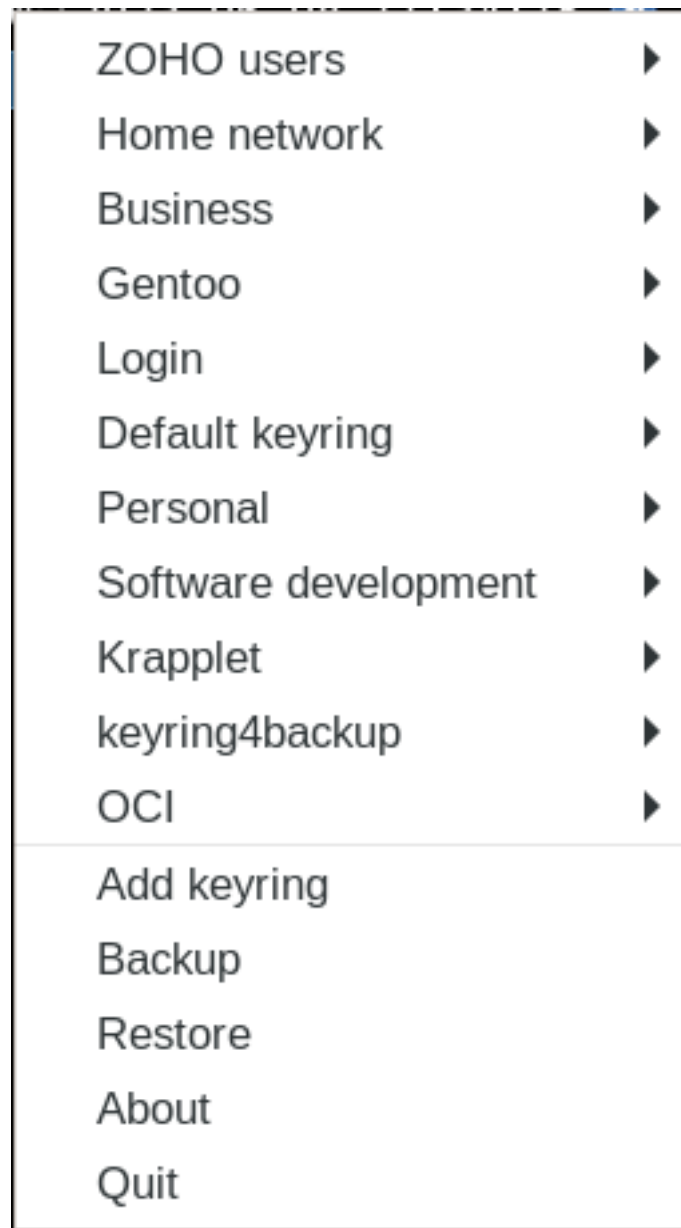
1.2.1 Krapplet in the systray

Once krapplet is loaded it will show itself as a keyring with some keys:



1.2.2 Main menu

The main menu is what you see once the systray icon is right-clicked:



1.2.3 Keyrings

Hovering over a keyring will show the keys in a keyring:

. image:

krapplet-submenu.png :align: center

1.2.4 Key window

The key window gives all information about a key:

- The keyring it belongs to. This combobox can be used to move a key to another keyring.
- The name of the key
- Some attributes: other information that might be relevant. There is a button to add more attributes. Clearing the attribute name removes the attribute from the key. If there is an attribute called URL (in capitals), then there will also be a launch button.
- The secret, aka password. Under the secret there is a little red/amber/green bar indicating the complexity of the password. There is a button to copy the secret to the clipboard, one to show the secret, and one to generate a new secret..

Edit key

Keyring

Software development

Key

Ubuntu Launchpad

Modified Sat 06 Feb 2021 05:07:26 PM

Attributes

URLhttps://www.launc

emailhfern@fernhout.ir

usernamehfern

AddLaunch

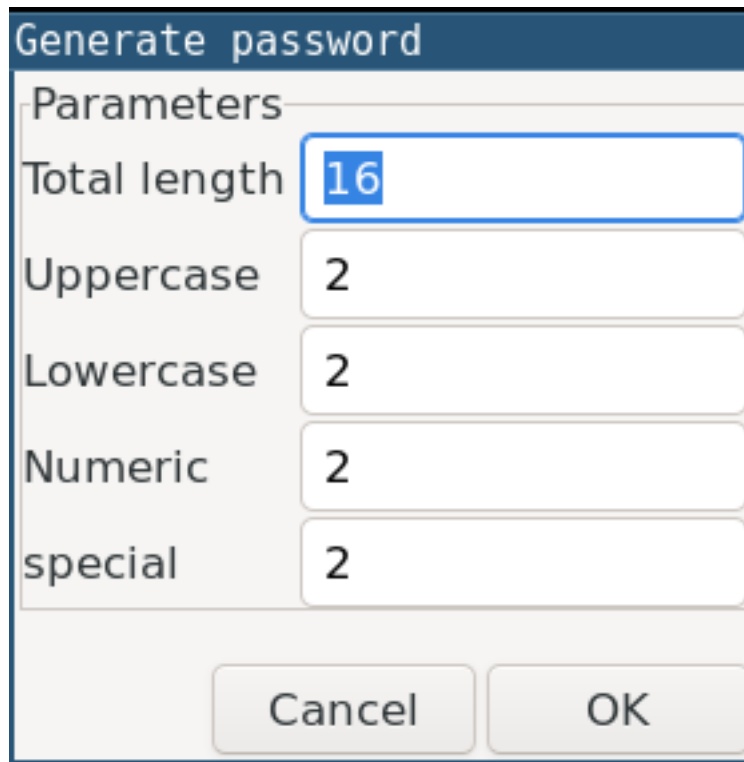
Secret

GenerateCopyShow

CancelDeleteOK

1.2.5 Generate password

Generation of passwords is based on the total requested length, and the minimum number of uppercase, lowercase, numeric and special characters:



The image shows a 'Generate password' dialog box. It has a title bar with the text 'Generate password'. Below the title bar is a section labeled 'Parameters'. Inside this section, there are five rows, each with a label and a text input field. The first row is 'Total length' with the value '16'. The second row is 'Uppercase' with the value '2'. The third row is 'Lowercase' with the value '2'. The fourth row is 'Numeric' with the value '2'. The fifth row is 'special' with the value '2'. At the bottom of the dialog box, there are two buttons: 'Cancel' and 'OK'.

Parameters	
Total length	16
Uppercase	2
Lowercase	2
Numeric	2
special	2

Cancel OK

1.3 Installation

Krapplet is developed in Python, so there is no compilation needed.

1.3.1 Installation sources

Krapplet is available:

- As source code on [Gitlab](#)
- From the Python Package Index [PyPI](#)
- From the Arch Linux' User Repository [AUR](#)
- From the Gentoo Linux' User Repository [GURU](#).

1.3.2 Software dependencies

Krapplet makes use of the following software:

- [Python](#) as the programming language, tested with versions 3.7, 3.8 and 3.9. Krapplet does not support Python 2.7.
- [GTK3](#) for the user interface. Developed against version 3.24.22. Please refer to [these instructions](#) for how to install GTK3 for your Linux distribution.
- [Secretstorage](#) as the Python API towards when using gnome-keyring as the storage provider. Secretstorage can be installed via [PyPI](#), but is also available as an operating system package in many Linux distributions. Secretstorage makes use of a Python package [cryptography](#), which in turn requires some operating system packages to be installed: see [here](#) for installation instructions.
- [Python-gnupg](#) when using the pass storage provider. Python-gnupg is a python wrapper around [GnuPG](#), which is used for the actual encryption of keys, and should be installed as well

1.3.3 Notes on installing using PyPI

PyPI is a very convenient way of installing Python packages and programs. However, be aware that:

- Dependencies may not always fully resolve, e.g. GTK3 cannot be installed from PyPI.
- Indirect dependencies that are not written in Python / are not available on PyPI won't be installed, e.g. GnuPG, or Gnome-Keyring.
- Anyone can put anything on PyPI: there is no control whether the right software is installed. There is for example also a package called gnupg on PyPI, which is a fork of python-gnupg, but which does not work for krapplet.

Krapplet has optional dependencies of which at least one need to be selected. When using krapplet with gnome-keyring then use:

```
# pip install krapplet[gnomekeyring]
```

When using krapplet with pass:

```
# pip install krapplet[pass]
```

Or when both storage providers are wanted:

```
# pip install krapplet[gnomekeyring,pass]
```

The recommendation is to install krapplet as an operating package; the person who packaged krapplet will have made sure all dependencies will be installed. Unfortunately not all operating systems provide a krapplet package, in which case installing from PyPI may be the next best option.

1.4 Usage

Once krapplet is started it will show itself as an icon in the system tray. When this icon is right clicked it will show a menu with the available keyrings, the option to add a keyring, and the backup, restore, about and quit options.

When hovering over a keyring it will give the option to unlock a keyring when it is locked or show the available keys, and options to add a key, remove the keyring, or lock the keyring.

Once a key has been selected it will show a small floating window to manipulate the key. It shows the key name as a first input field and when it is not a new key also the create and modified timestamps.

Next are the key attributes in name/value pairs. There is a button to add an additional attribute. A blank attribute name is interpreted as an instruction to remove that attribute. When an attribute name “URL” is present a launch button will be shown, which opens the system webbrowser for the listed URL when clicked.

The “secret” section contains the password, by default shown hidden. This section has buttons to generate a new password, copy the password to the system clipboard, and to show or hide the password.

The generation of a new password is based on five parameters:

- total length
- minimum number of uppercase, lowercase, numeric and special characters; use zero to exclude a category.

Special characters are defined as: !#\$%&()*+,-./:;<=>?@[\\]^_`{|}~

A colored bar may be seen under the secret: it is a password complexity indicator, which colors

- red when the password is too easy to crack, e.g. consisting of a mix of nine or less uppercase, lowercase, numeric, and special characters
- amber when there are ten or 11 characters of that same mix
- green when 12 or more.

The complexity indicator is based on its entropy.

From the main menu it is possible to backup or restore all keyrings in a password protected encrypted keyfile from/at a desired location.

1.5 Storage providers

Krapplet uses by default gnome-keyring as storage provider. As of version 0.3.0 krapplet also supports the storage architecture of [pass](#).

1.5.1 Gnome-keyring

Using gnome-keyring is default, but can be explicitly selected by passing this commandline parameter: `--storage gnome-keyring`.

How gnome-keyring stores secrets is described in [this gnome-keyring wiki page](#).

The upshot is that:

- There should be a password set for the first keyring to be unlocked, the login keyring
- That password should be the same as the password for the user’s Linux account

Desktop environments are capable of unlocking the login keyring as long as auto-login is not used.

1.5.2 Pass

Start krapplet with `--storage pass`: to select the pass compatible storage provider.

Note that the pass software code itself is not used; pass is written in bash, whereas krapplet is written in Python. Krapplet just mimics the way how pass stores passwords.

For the pass compatible storage provider there are the following optional command line switches:

- `--pgp-id <pgp-id>`: which pgp-id to use
- `--armor`: encode the password files in ASCII.

Pass and the krapplet pass compatible storage provider store their passwords in `${HOME}/.password-store`. Krapplet assumes this directory has been initialized before using the command `pass init`. This creates a file `.pgp-id` in this directory `${HOME}/.password-store`. In order to use the pass compatible storage provider `pgp` keys should be set up. Pass uses [PGP / RFC4880](#) encoded password files.

1.6 Disclaimers

This chapter contains some general information that users of krapplet should be aware of.

1.6.1 Notification icon

Notification icons are an officially deprecated method under GTK3. Use of the notification icon may or may not be abandoned in future releases.

1.6.2 Systray

Krapplet currently will not work properly on desktops that do not provide a systray.

1.6.3 Password generation

The generation of the password makes use of Python's random functions. These random functions rely on operating system features, typically the current time in fractions of seconds, or when available from a random device like `/dev/urandom` in Linux, which seems better. Even better would be if there is a hardware random number generator available and used by the operating system. How Python's random function works therefore depends on the Python interpreter, the system configuration, and the hardware.

The generated secret may not be good enough for real cryptographic purposes. Whether it is good enough for password generation is left to users of krapplet to decide.

1.6.4 Clipboard usage

Krapplet can copy secrets to the system clipboard, allowing a user to paste a secret in an input field, seemingly a secure way since bystanders cannot see what is being copied. However, please be weary of clipboard managers maintaining a clipboard history and therefore can reveal everything that has been copied to it, including passwords. Note that krapplet clears the clipboard after a key window is closed to prevent a password lingering in the clipboard, and getting accidentally pasted in a place where others might see it.

1.6.5 Swap

Memory constrained systems might swap krapplet out to a storage device. Encrypt your swap device or swap file to prevent hackers from harvesting secrets from it.

1.7 Troubleshooting

1.7.1 Error: could not embed in systray

- Make sure that there is a systray for krapplet to embed itself in. This behavior has been seen on GNOME-3 desktops.

1.7.2 Cannot see the krapplet icon

- Verify that there are krapplet.png files in `/usr/share/icons/hicolor/48x48/apps` and in `/usr/share/icons/hicolor/96x96/apps`
- Refresh the icon cache by running `gtk-update-icon-cache`

1.7.3 Login keyring not automatically unlocked

- Please check the official [gnome-keyring wiki](#), or your distribution's documentation, e.g. for [Arch Linux](#),

1.8 License

1.8.1 BSD 3-clause license

Copyright 2020-2021 Johannes Willem (Hans) Fernhout <hfern@fernhout.info> All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.8.2 Contact

Please contact Hans Fernhout via email for any license questions: hfern at fernhout dot info .

1.9 Revision history

1.9.1 Version 0.4.0 - 10 June 2023

New functionalities: - Confirm key delete - Add an option for backup and restore

1.9.2 Version 0.3.0/1 - 21 February 2021

New functionalities: - Secret validity added - Secret complexity indicator added - Functionality added to move secrets from one keyring to another - Auto-unlock keyrings when they are in the login keyring - Prevent deleting keyrings with keys attached to them - Generate a secret when a new key is created - Add ``pass https://www.passwordstore.org/`` as a storage provider

1.9.3 Version 0.2.0 - 17 January 2021

Updates include:

- Graceful exit on Ctrl-C: no longer Python crash log
- Clear clipboard after window close

1.9.4 Version 0.1.0 - 29 December 2020

Initial official release of krapplet, a password manager based on gnome-keyring.

Features of krapplet include:

- Systray applet: with a single right click you have the necessary login data immediately available
- keys are organized in keyrings, containing not only secrets (passwords), but also other related information (attributes)