

Менеджер паролей pass

Лабораторная работа №8

Козомазов Владимир Романович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.0.1	Теоретическое введение по теме “Менеджер паролей pass”	7
3.0.2	1. Что такое pass?	7
3.0.3	2. Принципы работы pass	8
3.0.4	3. Преимущества pass	9
3.0.5	4. Основные команды pass	10
3.0.6	5. Пример использования	11
4	Выполнение лабораторной работы	13
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	Установка менеджера паролей	13
4.2	Завершение установки менеджера паролей	13
4.3	Просмотр списка ключей	13
4.4	Инициализация хранилища	13
4.5	Создание структуры	14
4.6	Ручное выкладывание изменений	14
4.7	Проверка статуса синхронизаций	14
4.8	Добавление нового пароля	14
4.9	Установка дополнительного программного обеспечения	14
4.10	Установка дополнительных шрифтов	15

Список таблиц

1 Цель работы

Цель установки менеджера паролей — обеспечить безопасное, удобное и эффективное управление учетными данными. Менеджер паролей помогает: - Создавать и использовать сложные уникальные пароли. - Защищать данные от утечек и атак. - Упрощать процесс входа на сайты и в приложения. - Организовывать и синхронизировать пароли между устройствами.

2 Задание

- Установка менеджера паролей *pass*
- Установка дополнительного программного обеспечения
- Создать собственный репозиторий с помощью утилит
- Изучить *chezmoi*

3 Теоретическое введение

3.0.1 Теоретическое введение по теме “Менеджер паролей pass”

Менеджер паролей — это инструмент, предназначенный для безопасного хранения, генерации и управления паролями и другими конфиденциальными данными. В современном мире, где количество онлайн-аккаунтов у одного пользователя может исчисляться десятками, использование менеджера паролей становится необходимым для обеспечения безопасности и удобства.

Одним из таких инструментов является **pass** — простой, но мощный менеджер паролей, основанный на стандартных Unix-утилитах и принципах открытого исходного кода. В этом теоретическом введении мы рассмотрим основные концепции, принципы работы и преимущества использования менеджера паролей **pass**.

3.0.2 1. Что такое pass?

pass — это консольный менеджер паролей, который использует **GPG** (GNU Privacy Guard) для шифрования и хранения паролей. Он был создан в 2006 году Джейсоном Доненфельдом (Jason A. Donenfeld) и с тех пор завоевал популярность благодаря своей простоте, безопасности и интеграции с Unix-подобными системами.

Основные особенности **pass**: - Хранение паролей в зашифрованных файлах.
- Использование GPG для шифрования. - Интеграция с Git для синхронизации

паролей между устройствами. - Поддержка плагинов для расширения функциональности.

3.0.3 2. Принципы работы pass

3.0.3.1 2.1. Хранение паролей

- Пароли хранятся в виде зашифрованных файлов в директории `~/.password-store/`.
- Каждый пароль — это отдельный файл, зашифрованный с использованием GPG.
- Имена файлов соответствуют названиям сервисов или аккаунтов, например:

`~/.password-store/example.com/user.gpg`

`~/.password-store/github.com/username.gpg`

3.0.3.2 2.2. Шифрование

- **pass** использует GPG для шифрования данных. Каждый файл с паролем зашифрован с использованием открытого ключа GPG пользователя.
- Для расшифровки используется закрытый ключ GPG, что обеспечивает высокий уровень безопасности.

3.0.3.3 2.3. Организация паролей

- Пароли могут быть организованы в иерархическую структуру директорий. Например:


```
~/.password-store/  
├─ work/  
│   ├── vpn.gpg  
│   └─ email.gpg  
├─ personal/  
│   ├── bank.gpg  
│   └─ social.gpg
```

3.0.3.4 2.4. Синхронизация с Git

- **pass** поддерживает интеграцию с Git, что позволяет синхронизировать пароли между несколькими устройствами.
 - Все изменения в хранилище паролей автоматически фиксируются в Git, что упрощает отслеживание истории изменений.
-

3.0.4 3. Преимущества pass

3.0.4.1 3.1. Безопасность

- Использование GPG обеспечивает высокий уровень шифрования.
- Пароли хранятся локально, что снижает риск утечки данных через облачные сервисы.

3.0.4.2 3.2. Простота

- **pass** использует стандартные Unix-утилиты (GPG, Git), что делает его легким в установке и использовании.
- Интерфейс командной строки интуитивно понятен и удобен для опытных пользователей.

3.0.4.3 3.3. Гибкость

- Поддержка плагинов позволяет расширять функциональность **pass** (например, интеграция с браузерами, генерация одноразовых паролей и т.д.).
- Возможность организации паролей в иерархическую структуру.

3.0.4.4 3.4. Открытый исходный код

- **pass** распространяется под лицензией GPL, что позволяет пользователям проверять и модифицировать код.
-

3.0.5 4. Основные команды pass

- **Инициализация хранилища:**

```
pass init <gpg-key-id>
```

Создает новое хранилище паролей, зашифрованное с использованием указанного GPG-ключа.

- **Добавление пароля:**

```
pass insert <путь/к/паролю>
```

Добавляет новый пароль в хранилище.

- **Просмотр пароля:**

```
pass <путь/к/паролю>
```

Отображает пароль для указанного сервиса.

- **Генерация пароля:**

```
pass generate <путь/к/паролю> <длина>
```

Генерирует случайный пароль указанной длины.

- **Синхронизация с Git:**

```
pass git init
pass git remote add origin <репозиторий>
pass git push
```

Инициализирует Git-репозиторий и синхронизирует пароли с удаленным хранилищем.

3.0.6 5. Пример использования

1. Инициализация хранилища:

```
pass init 1234ABCD
```

(где 1234ABCD — идентификатор GPG-ключа).

2. Добавление пароля:

```
pass insert example.com/user
```

3. Просмотр пароля:

```
pass example.com/user
```

4. Генерация пароля:

```
pass generate example.com/newuser 20
```

5. Синхронизация с Git:

```
pass git init
pass git remote add origin git@example.com:password-store.git
pass git push
```

pass — это мощный и гибкий менеджер паролей, который сочетает в себе простоту использования, высокий уровень безопасности и интеграцию с Unix-экосистемой. Он идеально подходит для пользователей, которые предпочитают консольные инструменты и хотят иметь полный контроль над своими данными. Благодаря использованию GPG и Git, **pass** обеспечивает надежное шифрование и удобную синхронизацию паролей между устройствами.

4 Выполнение лабораторной работы

Установил менеджер паролей pass с помощью команды `sudo dnf install pass` `pass-otp` (рис. 4.1).

```
Kozomazov@fedora: ~$ sudo dnf install pass pass-otp
[sudo] пароль для vkozomazov:
Пополнение кэш паз.
[sudo] пароль для vkozomazov:
Updating and loading repositories:
Repositories loaded.
Name: 'pass-1.7.4-11.fc41.noarch' уже установлен.
Name: 'pass-otp-1.2.0-15.fc41.noarch' уже установлен.
Nothing to do.
Kozomazov@fedora: ~$
```

Рис. 4.1: Установка менеджера паролей

Завершил установку менеджера паролей командой `sudo dnf install gopass` (рис. 4.2).

```
Kozomazov@fedora: ~$ sudo dnf install gopass
Updating and loading repositories:
Repositories loaded.
Name: 'gopass-1.15.15-2.fc41.x86_64' уже установлен.
Nothing to do.
Kozomazov@fedora: ~$
```

Рис. 4.2: Завершение установки менеджера паролей

Просмотрел список `gpg` ключей при помощи команды `gpg --list-secret-keys` (рис. 4.3).

```
Kozomazov@fedora: ~$ gpg --list-secret-keys
[keyboard]
-----
sec   Ts4095 2025-03-14 [SC]
      4938B838E2482245D91F62F84FFBD2FEE4272D8C
uid   [ abcomerco ] Vladimir Kozomazov <voffkoc@gmail.com>
sub   Ts4095 2025-03-14 [E]
Kozomazov@fedora: ~$
```

Рис. 4.3: Просмотр списка ключей

Инициализировал хранилище, написав команду `'pass init` (рис. 4.4).

```
Kozomazov@fedora: ~$ pass init 4938B838E2482245D91F62F84FFBD2FEE4272D8C
Password store initialized for 4938B838E2482245D91F62F84FFBD2FEE4272D8C
```

Рис. 4.4: Инициализация хранилища

Создал структуру с git командой `pass git init` (рис. 4.5).

```
kozomazov@fedora:~$ pass git init
Перициализирован существующий репозиторий Git в /home/kozomazov/.password-store/.git/
kozomazov@fedora:~$
```

Рис. 4.5: Создание структуры

Синхронизировался с git командами `pass git pull`, `pass git push`
Вручную закоммитил и выложил изменения командами (рис. 4.6).

```
kozomazov@fedora:~$ cd ~/.password-store/
kozomazov@fedora:~/.password-store$ git add .
kozomazov@fedora:~/.password-store$ git commit -am 'edit manually'
[main 6b7c4: 1 commit]
kozomazov@fedora:~/.password-store$ git push
Everything's up-to-date.
```

Рис. 4.6: Ручное выкладывание изменений

Проверил статус синхронизации командой `pass git status` (рис. 4.7).

```
kozomazov@fedora:~/.password-store$ pass git status
On branch master
Your branch is up-to-date with the upstream 'master'.
nothing to commit, working tree clean
```

Рис. 4.7: Проверка статуса синхронизаций

Добавил новый пароль командой `pass insert [OPTIONAL DIR]/[FILENAME]` (рис. 4.8).

```
kozomazov@fedora:~/.password-store$ pass insert tests/test1
An entry already exists for tests/test1. Overwrite it? (y/N) y
Enter password for tests/test1:
Retype password for tests/test1:
[main a41f640] Add given password for tests/test1 to store.
1 file changed, 0 insertions(+), 0 deletions(-)
```

Рис. 4.8: Добавление нового пароля

Отобразил пароль для указанного имени файла `pass [OPTIONAL DIR]/[FILENAME]`
Установил дополнительное программное обеспечение (рис. 4.9).

```
light \
fuzzel \
swaylock \
kitty \
waybar \
wl-clipboard \
mpv \
grim \
slurp

[sudo] пароль для kozomazov:
Updating and loading repositories:
Repositories loaded.
Package 'dunst-1.12.1-1.fc41.x86_64' уже установлен.
Package 'fontawesome-fonts-10.2.3-1.fc41.noarch' уже установлен.
Package 'powerline-fonts-2.8.4-1.fc41.noarch' уже установлен.
Package 'light-1.2.2-1.fc41.x86_64' уже установлен.
Package 'fuzzel-1.11.1-1.fc41.x86_64' уже установлен.
Package 'swaylock-1.8.0-1.fc41.x86_64' уже установлен.
Package 'kitty-0.19.1-1.fc41.x86_64' уже установлен.
Package 'waybar-0.11.0-1.fc41.x86_64' уже установлен.
Package 'swaybg-1.2.1-1.fc41.x86_64' уже установлен.
Package 'wl-clipboard-2.2.1-1.fc41.x86_64' уже установлен.
Package 'mpv-0.39.0-1.fc41.x86_64' уже установлен.
Package 'grim-1.4.1-1.fc41.x86_64' уже установлен.
Package 'slurp-1.5.0-1.fc41.x86_64' уже установлен.

Nothing to do.
```

Рис. 4.9: Установка дополнительного программного обеспечения

```
usevka-term-ss80-fonts.monoarch Monospace, Liberation Mono Style  
usevka-term-s87-fonts.monoarch Monospace, Monaco Style  
usevka-term-ss98-fonts.monoarch Monospace, Pragmata Pro Style  
usevka-term-u809-fonts.monoarch Monospace, Source Code Pro Style  
usevka-term-u816-fonts.monoarch Monospace, Envy Code K Style  
usevka-term-ts11-fonts.monoarch Monospace, X Windows Fixed Style  
usevka-term-tsl2-fonts.monoarch Monospace, Ubuntu Mono Style  
usevka-term-us13-fonts.monoarch Monospace, Lucida Style  
usevka-term-ts14-fonts.monoarch Monospace, JetBrains Mono Style  
usevka-term-us15-fonts.monoarch Monospace, IBM Plex Mono Style  
usevka-term-us16-fonts.monoarch Monospace, PT Mono Style  
usevka-term-ts17-fonts.monoarch Monospace, Recursive Mono Style  
usevka-term-us18-fonts.monoarch Monospace, Input Mono Style
```

\$ sudo apt install usevka-aile-fonts usevka-curly-fonts usevka-zlab-fonts usevka-etolele-fonts usevka-term-fonts

Updating and loading repositories:

```
# Repositories loaded.  
# Query "isevka-fonts-33.0.1-1.fc41.noarch" уже установлен.  
# Query "isevka-aile-fonts-33.0.1-1.fc41.noarch" уже установлен.  
# Query "isevka-curly-fonts-33.0.1-1.fc41.noarch" уже установлен.  
# Query "isevka-zlab-fonts-33.0.1-1.fc41.noarch" уже установлен.  
# Query "isevka-etolele-fonts-33.0.1-1.fc41.noarch" уже установлен.  
# Query "isevka-term-fonts-33.0.1-1.fc41.noarch" уже установлен.
```

Nothing to do.

5 Выводы

pass — это мощный и гибкий менеджер паролей, который сочетает в себе простоту использования, высокий уровень безопасности и интеграцию с Unix-экосистемой. Он идеально подходит для пользователей, которые предпочитают консольные инструменты и хотят иметь полный контроль над своими данными. Благодаря использованию GPG и Git, **pass** обеспечивает надежное шифрование и удобную синхронизацию паролей между устройствами.

Список литературы