Менеджер паролей pass

Лабораторная работа №5

Козомазов Владимир Романович

Содержание

Список иллюстраций

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).

1. Добавление пароля:

* pass insert example.com/user

1. Просмотр пароля:

* pass example.com/user

1. Генерация пароля:

* pass generate example.com/newuser 20

1. 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. 1).



Рис. 1: Установка менеджера паролей

Завершил установку менеджера паролей командой sudo dnf install gopass (рис. 2).



Рис. 2: Завершение установки менеджера паролей

Просмотрел список gpg ключей при помощи команды gpg --list-secret-keys (рис. 3).



Рис. 3: Просмотр списка ключей

Иницилизировал хранилище, написав команду `pass init (рис. 4).

Рис. 4: Иницилизация хранилища

Рис. 4: Иницилизация хранилища

Создал структуру с git командой pass git init (рис. 5).

Рис. 5: Создание структуры

Рис. 5: Создание структуры

Синхронизировался с git командами pass git pull, pass git push

Вручную закоммитил и выложил изменения командами (рис. 6).



Рис. 6: Ручное выкладывание изменений

Проверил статус синхронизации командой pass git status (рис. 7).

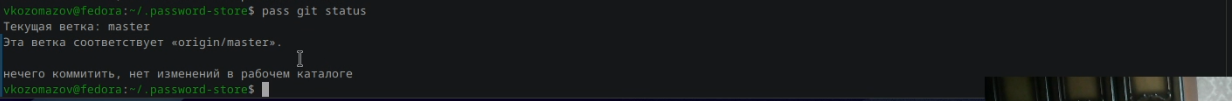


Рис. 7: Проверка статуса синхронизаций

Добавил новый пароль командой pass insert [OPTIONAL DIR]/[FILENAME] (рис. 8).

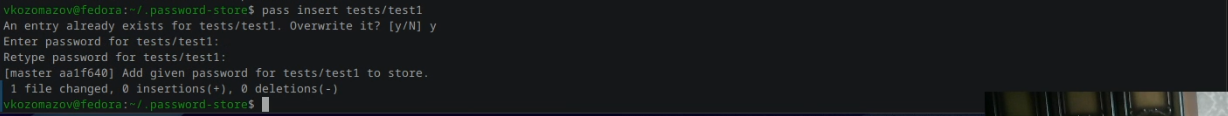


Рис. 8: Добавление нового пароля

Отобразил пароль для указанного имени файла pass [OPTIONAL DIR]/[FILENAME]

Установил дополнительное программное обеспечение (рис. 9).

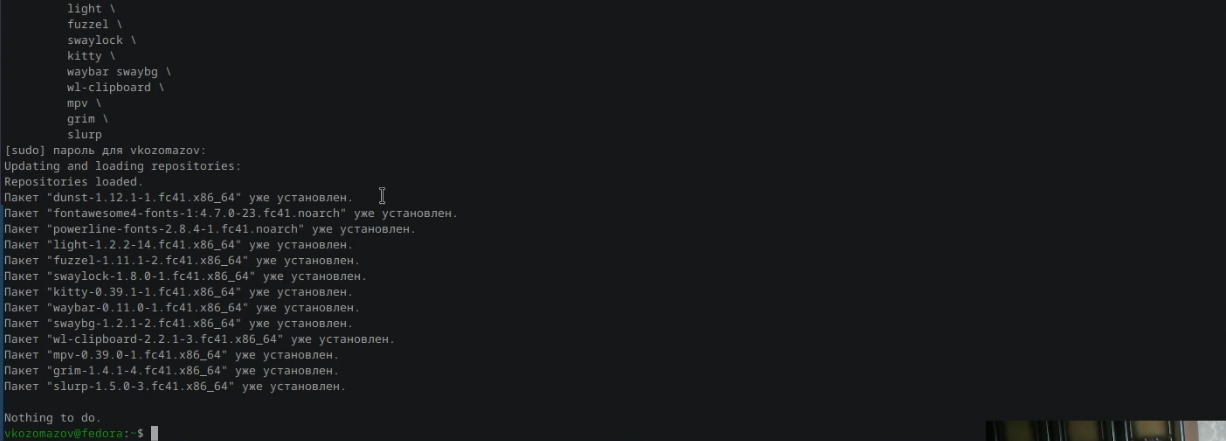


Рис. 9: Установка дополнительного программного обеспечения

Установил дополнительно шрифты командами sudo dnf copr enable peterwu/iosevka, sudo dnf search iosevka, sudo dnf install iosevka-fonts iosevka-aile-fonts iosevka-curly-fonts iosevka-slab-fonts iosevka-etoile-fonts iosevka-term-fonts (рис. 10).



Рис. 10: Установка дополнительных шрифтов

Установил бинарный файл командой sh -c "$(wget -qO- chezmoi.io/get)"

Создал собственный репозиторий при помощи утилит на основе шаблона

Подключил репозиторий к своей системе

Использовал chezmoi на нескольких машинах

Настроил новую машину с помощью одной команды

Включил функцию автоматической фиксации и отправлении изменений в репозиторий

# 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