

Отчет по лабораторной работе №5

Настройка рабочей среды.

Дагделен Зейнап Реджеповна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Менеджер паролей pass	7
3.2	Основные свойства	7
3.3	Структура базы паролей	7
3.4	Семантическая структура базы паролей	8
4	Выполнение лабораторной работы	9
4.1	Установка	9
4.2	Настройка	9
5	Выводы	18
	Список литературы	19

Список иллюстраций

4.1	Установка pass	9
4.2	Установка gorass	9
4.3	Просмотр списка ключей и инициализация хранилища	10
4.4	Создание структуры git	10
4.5	Создание репозитория	10
4.6	Задание адреса репозитория на хостинге	11

Список таблиц

1 Цель работы

Научиться настраивать рабочую среду

2 Задание

- 1) Менеджер паролей pass -Установка -Настройка -Настройка интерфейса с браузером -Сохранение пароля
- 2) Управление файлами конфигурации
- 3) Дополнительное программное обеспечение -Установка -Создание собственного репозитория с помощью утилит -Подключение репозитория к своей системе -Использование chezmoi на нескольких машинах -Настройка новой машины с помощью одной команды -Ежедневные операции с chezmoi

3 Теоретическое введение

3.1 Менеджер паролей pass

- Менеджер паролей pass — программа, сделанная в рамках идеологии Unix.
- Также носит название стандартного менеджера паролей для Unix (The standard Unix password manager).

3.2 Основные свойства

- Данные хранятся в файловой системе в виде каталогов и файлов.
- Файлы шифруются с помощью GPG-ключа.

3.3 Структура базы паролей

- Структура базы может быть произвольной, если Вы собираетесь использовать её напрямую, без промежуточного программного обеспечения. Тогда семантику структуры базы данных Вы держите в своей голове.
- Если же необходимо использовать дополнительное программное обеспечение, необходимо семантику заложить в структуру базы паролей.

3.4 Семантическая структура базы паролей

Рассмотрим пользователя user в домене example.com, порт 22. Отсутствие имени пользователя или порта в имени файла означает, что любое имя пользователя и порт будут совпадать:

example.com.pgp

Соответствующее имя пользователя может быть именем файла внутри каталога, имя которого совпадает с хостом. Это полезно, если в базе есть пароли для нескольких пользователей на одном хосте:

example.com/user.pgp

Имя пользователя также может быть записано в виде префикса, отделенного от хоста знаком @:

user@example.com.pgp

Соответствующий порт может быть указан после хоста, отделённый двоеточием (:):

example.com:22.pgp example.com:22/user.pgp user@example.com:22.pgp

Эти все записи могут быть расположены в произвольных каталогах, задающих Вашу собственную иерархию.

4 Выполнение лабораторной работы

4.1 Установка

Устанавливаю pass и gopass на Ubuntu (рис. 4.1 - рис. 4.2).

```
zrdagdelen@zrdagdelen-MacBookPro:~$ sudo apt install pass
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  qrencode tree
Предлагаемые пакеты:
  libxml-simple-perl python
Следующие NOBЫE пакеты будут установлены:
  pass qrencode tree
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 108 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 314 кВ.
Хотите продолжить? [Д/н] ☐
```

Рис. 4.1: Установка pass

```
zrdagdelen@zrdagdelen-MacBookPro:~$ sudo apt install gopass
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие NOBЫE пакеты будут установлены:
  gopass
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 821 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 2 204 кВ.
Пол:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 gopass amd64 1.5.0-1 [821 кВ]
Получено 821 кВ за 0с (3 993 кВ/с)
☐
```

Рис. 4.2: Установка gopass

4.2 Настройка

Просмотрю список ключей GPG (gpg --list-secret-keys) и инициализируем хранилище (рис. 4.3), перейду в каталог work с помощью cd.

```

zrdagdelen@zrdagdelen-MacBookPro:~$ gpg --list-secret-keys
/home/zrdagdelen/.gnupg/pubring.kbx
-----
sec   rsa4096 2024-03-08 [SC]
      3337BFC7236BF43F030DED8A4D27B76D0B80DE3C
uid    [ абсолютно ] Zeynap Dagdelen <zrdagdelenn@gmail.com>
ssb    rsa4096 2024-03-08 [E]

zrdagdelen@zrdagdelen-MacBookPro:~$ pass init zrdagdelenn@gmail.com
mkdir: создан каталог '/home/zrdagdelen/.password-store/'
Password store initialized for zrdagdelenn@gmail.com
zrdagdelen@zrdagdelen-MacBookPro:~$ cd work

```

Рис. 4.3: Просмотр списка ключей и инициализация хранилища

Синхронизация с git. Создам структуру git (pass git init)(рис. 4.4)

```

Каталога
zrdagdelen@zrdagdelen-MacBookPro:~/work$ pass git init
Инициализирован пустой репозиторий Git в /home/zrdagdelen/.password-store/.git/
[master (корневой коммит) 94e2a74] Add current contents of password store.
1 file changed, 1 insertion(+)
create mode 100644 .gpg-id
[master 9ec99ca] Configure git repository for gpg file diff.
1 file changed, 1 insertion(+)
create mode 100644 .gitattributes
zrdagdelen@zrdagdelen-MacBookPro:~/work$

```

Рис. 4.4: Создание структуры git

Также можно задать адрес репозитория на хостинге (репозиторий необходимо предварительно создать)(рис. 4.5 - рис. 4.6).

Create a new repository

A repository contains all project files, including the revision history. Already have a p repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk ().*

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner *

 zrdagdelen ▾

Repository name *

git_repo_for_lab5

✔ git_repo_for_lab5 is available.

Great repository names are short and memorable. Need inspiration? How about [studious-telegram](#) ?

Рис. 4.5: Создание репозитория

```
zrdagdelen@zrdagdelen-MacBookPro:~/work$ pass git remote add origin git@github.com:zrdagdelen/git_repo_for_lab5.git
zrdagdelen@zrdagdelen-MacBookPro:~/work$ pass git pull
```

Рис. 4.6: Задание адреса репозитория на хостинге

Для синхронизации выполняю следующую команду

```
pass git pull
pass git push
```

Прямые изменения

Следует заметить, что отслеживаются только изменения, сделанные через сам `gorpass` (или `pass`).

Если изменения сделаны непосредственно на файловой системе, необходимо вручную закоммитить и выложить изменения:

```
cd ~/.password-store/
git add .
git commit -am 'edit manually'
git push
```

Проверить статус синхронизации командой

```
pass git status
```

Настройка интерфейса с браузером

Для взаимодействия с браузером используется интерфейс `native messaging`.

Поэтому кроме плагина к браузеру устанавливается программа, обеспечивающая интерфейс

Плагин `browserpass`

Репозиторий: <https://github.com/browserpass/browserpass-extension>

Плагин для браузера

Плагин для Firefox: <https://addons.mozilla.org/en-US/firefox/addon/browserpass/>

Плагин для Chrome/Chromium: <https://chrome.google.com/webstore/detail/browserpass/naepdomgkenhinolocfifgehiddafch>.

Интерфейс для взаимодействия с браузером (native messaging)

Репозиторий: <https://github.com/browserpass/browserpass-native>

Gentoo:

```
emerge www-plugins/browserpass
```

Fedora

```
dnf copr enable maximbaz/browserpass
```

```
dnf install browserpass
```

Сохранение пароля

Добавить новый пароль

Выполните:

```
pass insert [OPTIONAL DIR]/[FILENAME]
```

OPTIONAL DIR: необязательное имя каталога, определяющее файловую структуру

FILENAME: имя файла, который будет использоваться для хранения пароля.

Отобразите пароль для указанного имени файла:

```
pass [OPTIONAL DIR]/[FILENAME]
```

Замените существующий пароль:

```
pass generate --in-place FILENAME
```

Управление файлами конфигурации

Дополнительное программное обеспечение

Установите дополнительное программное обеспечение:

```
sudo dnf -y install \  
    dunst \  
    fontawesome-fonts \  
    powerline-fonts \  
    light \  
    fuzzel \  
    swaylock \  
    kitty \  
    waybar swaybg \  
    wl-clipboard \  
    mpv \  
    grim \  
    slurp
```

Установите шрифты:

```
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
```

Установка

Установка бинарного файла. Скрипт определяет архитектуру процессора и операционную систему.

с помощью wget:

```
sh -c "$(wget -qO- chezmoi.io/get)"
```

Создание собственного репозитория с помощью утилит

Будем использовать утилиты командной строки для работы с github.

Создадим свой репозиторий для конфигурационных файлов на основе шаблона:

```
gh repo create dotfiles --template="yamadharma/dotfiles-template" --private
```

Подключение репозитория к своей системе

Инициализируйте chezmoi с вашим репозиторием dotfiles:

```
chezmoi init git@github.com:<username>/dotfiles.git
```

Проверьте, какие изменения внесёт chezmoi в домашний каталог, запустив:

```
chezmoi diff
```

Если вас устраивают изменения, внесённые chezmoi, запустите:

```
chezmoi apply -v
```

Использование chezmoi на нескольких машинах

На второй машине инициализируйте chezmoi с вашим репозиторием dotfiles:

```
chezmoi init https://github.com/<username>/dotfiles.git
```

Или через ssh:

```
chezmoi init git@github.com:<username>/dotfiles.git
```

Проверьте, какие изменения внесёт chezmoi в домашний каталог, запустив:

```
chezmoi diff
```

Если вас устраивают изменения, внесённые chezmoi, запустите:

```
chezmoi apply -v
```

Если вас не устраивают изменения в файле, отредактируйте его с помощью:

```
chezmoi edit file_name
```

Также можно вызвать инструмент слияния, чтобы объединить изменения между текущим

```
chezmoi merge file_name
```

При существующем каталоге chezmoi можно получить и применить последние изменения

```
chezmoi update -v
```

Настройка новой машины с помощью одной команды

Можно установить свои dotfiles на новый компьютер с помощью одной команды:

```
chezmoi init --apply https://github.com/<username>/dotfiles.git
```

Через ssh:

```
chezmoi init --apply git@github.com:<username>/dotfiles.git
```

Ежедневные операции с chezmoi

Извлеките последние изменения из репозитория и примените их

Можно извлечь изменения из репозитория и применить их одной командой:

```
chezmoi update
```

Это запускается `git pull --autostash --rebase` в вашем исходном каталоге, а за

Извлеките последние изменения из своего репозитория и посмотрите, что изменится,

Выполните:

```
chezmoi git pull -- --autostash --rebase && chezmoi diff
```

Это запускается `git pull --autostash --rebase` в вашем исходном каталоге, а ch

Если вы довольны изменениями, вы можете применить их:

```
chezmoi apply
```

Автоматически фиксируйте и отправляйте изменения в репозиторий

Можно автоматически фиксировать и отправлять изменения в исходный каталог в репозиторий. Эта функция отключена по умолчанию.

Чтобы включить её, добавьте в файл конфигурации `~/.config/chezmoi/chezmoi.toml`:

```
[git]
```

```
    autoCommit = true
```

```
    autoPush = true
```

Всякий раз, когда в исходный каталог вносятся изменения, `chezmoi` фиксирует их. Будьте осторожны при использовании `autoPush`. Если ваш репозиторий `dotfiles` является публичным, то вы можете не захотеть, чтобы изменения в исходный каталог автоматически отправлялись в репозиторий.

5 Выводы

Я настроила рабочую среду

Список литературы

Операционные системы