

Лабораторная работа №5

Настройка рабочей среды

Буллер Татьяна Александровна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Установка необходимого ПО.	5
2.2	Сохранение пароля	7
2.3	Подключение репозитория к своей системе	8
2.4	Ежедневные операции с chezmoi	9
3	Выводы	11

Список иллюстраций

2.1	Установка pass	5
2.2	Установка gorpass	5
2.3	Начало листа grg ключей	6
2.4	Инициализация хранилища	6
2.5	Инициализация гит	6
2.6	Инициализация гит	6
2.7	Добавление в гитхаб	7
2.8	Проверка статуса изменений	7
2.9	Создание пароля	7
2.10	Просмотр существующего пароля и генерация нового	8
2.11	Установка chezmoi	8
2.12	Создание репозитория	8
2.13	Инициализация chezmoi	9
2.14	Часть вывода изменений	9
2.15	Простые операции с chezmoi	10
2.16	Редактирование файла конфигурации	10

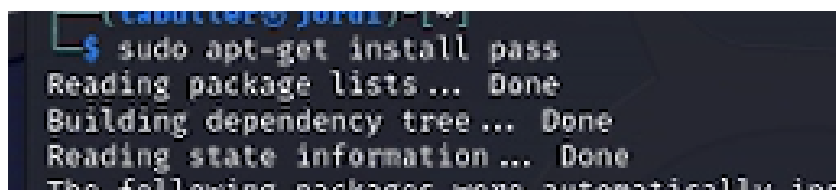
1 Цель работы

Получение навыков правильной работы с менеджером паролей pass и использования chezmoi для управления файлами конфигурации домашнего каталога пользователя.

2 Выполнение лабораторной работы

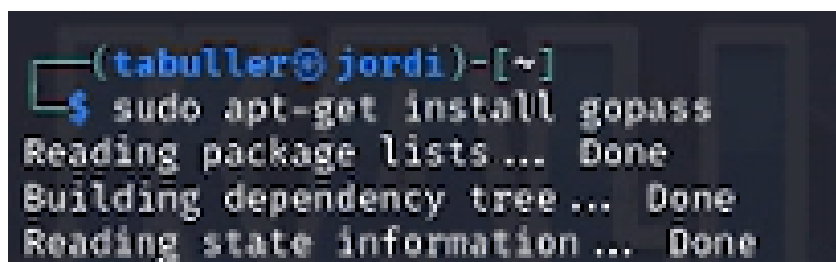
2.1 Установка необходимого ПО.

Необходимо установить pass и gopass, в дистрибутивах Debian используем для этого apt-get.



```
(tabuller@jordi)-[~]  
$ sudo apt-get install pass  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed by  
the system:
```

Рис. 2.1: Установка pass



```
(tabuller@jordi)-[~]  
$ sudo apt-get install gopass  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

Рис. 2.2: Установка gopass

После установки менеджера паролей проверим существующие на устройстве GPG ключи:

```
(tabuller@jordi)-[~]
$ gpg --list-secret-keys
/home/buller/.gnupg/pubring.kbx

sec     rsa4096 2024-02-17 [SC]
7232ACC0E976CBEE1CBAD10DDE753361AA440EE6
```

Рис. 2.3: Начало листа gpg ключей

Ключи на устройстве уже есть, создавать дополнительные не нужно. Переходим к следующему шагу - инициализации хранилища.

```
(tabuller@jordi)-[~]
$ pass init 7232ACC0E976CBEE1CBAD10DDE753361AA440EE6
mkdir: created directory '/home/buller/.password-store/'
Password store initialized for 7232ACC0E976CBEE1CBAD10DDE753361AA440EE6
```

Рис. 2.4: Инициализация хранилища

Для инициализации можем использовать почту либо ключ GPG. Инициализируем по ключу.

```
(tabuller@jordi)-[~]
$ pass git init
Initialized empty Git repository in /home/buller/.password-store/.git/
[master (root-commit) b525ee4] Add current contents of password store.
1 file changed, 1 insertion(+)
create mode 100644 .gpg-id
[master d2c825a] Configure git repository for gpg file diff.
1 file changed, 1 insertion(+)
create mode 100644 .gitattributes
```

Рис. 2.5: Инициализация гит

Связываем pass с git. В созданный репозиторий добавляется содержание текущего хранилища паролей.

```
(tabuller@jordi)-[~]
$ pass git remote add origin git@github.com:sarykush/study_2023-2024_os-intro.git
```

Рис. 2.6: Инициализация гит

Зададим также адрес репозитория на хостинге - рабочего репозитория предмета. Далее добавляем изменения (т.к. хранилище пусто, изменений пока что

нет).

```
(tabuller@jordi)-[~]
$ pass git pull
remote: Enumerating objects: 167, done.
remote: Counting objects: 100% (167/167), done.
remote: Compressing objects: 100% (131/131), done.
remote: Total 167 (delta 37), reused 145 (delta 28), pack-reused 0
Receiving objects: 100% (167/167), 19.69 MiB | 3.17 MiB/s, done.
Resolving deltas: 100% (37/37), done.
From github.com:sarykush/study_2023-2024_os-intro
* [new branch]      develop    -> origin/develop
* [new branch]      master     -> origin/master
* [new tag]         v1.0.0     -> v1.0.0
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.
```

Рис. 2.7: Добавление в гитхаб

```
(tabuller@jordi)-[~]
$ pass git status
On branch master
nothing to commit, working tree clean
```

Рис. 2.8: Проверка статуса изменений

Рабочее дерево чисто, добавлять нечего - можно продолжать работу.

2.2 Сохранение пароля

Создадим пароль. Без указания директории файл сохранится в директорию по умолчанию. Программа просит ввести пароль и повторить его:

```
(tabuller@jordi)-[~]
$ pass insert /SarYkuSHHpaS
Enter password for /SarYkuSHHpaS:
Retype password for /SarYkuSHHpaS:
[master 506db68] Add given password for /SarYkuSHHpaS to store.
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 SarYkuSHHpaS.gpg
```

Рис. 2.9: Создание пароля

Для просмотра пароля достаточно просто использовать `pass` с именем файла пароля без дополнительных команд. Просматриваем пароль, затем генерируем на его месте новый случайный с помощью `pass generate --in-place`

```
(tabuller@jordi)-[~]  
$ pass /SarYkuSHHpaS  
579315  
  
(tabuller@jordi)-[~]  
$ pass generate --in-place SarYkuSHHpaS  
[master 3d91b3b] Replace generated password for SarYkuSHHpaS.  
1 file changed, 0 insertions(+), 0 deletions(-)  
The generated password for SarYkuSHHpaS is:
```

Рис. 2.10: Просмотр существующего пароля и генерация нового

2.3 Подключение репозитория к своей системе

Для подключения репозитория понадобится утилита `chezmoi`. Устанавливаем ее:

```
(tabuller@jordi)-[~]  
$ sh -c "$(wget -qO- chezmoi.io/get)"  
info found version 2.47.1 for latest/linux/amd64  
info installed ./bin/chezmoi
```

Рис. 2.11: Установка `chezmoi`

После установки необходимого ПО создаем репозиторий `dotfiles` из шаблона:

```
(tabuller@jordi)-[~]  
$ gh repo create dotfiles --template="yamadharmadotfiles-template" --private  
v Created repository sarykush/dotfiles on GitHub  
https://github.com/sarykush/dotfiles
```

Рис. 2.12: Создание репозитория

Следующим шагом необходимо провести инициализацию ранее установленной утилиты с созданным репозиторием. Используем для этого команду `chezmoi init`:


```
(tabuller@jordi)-[~]
$ ./bin/chezmoi init git@github.com:sarykush/dotfiles.git
Cloning into '/home/buller/.local/share/chezmoi' ...
remote: Enumerating objects: 100, done.
remote: Counting objects: 100% (100/100), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 100 (delta 0), reused 99 (delta 0), pack-reused 0
Receiving objects: 100% (100/100), 77.86 KiB | 866.00 KiB/s, done.
```

Рис. 2.13: Инициализация chezmoi

После инициализации проверяем, какие изменения предлагаются к внесению. Вывод команды довольно большой, включает в себя закомментированные строки пояснений и предлагаемые изменения. Соглашаемся с изменениями, введя команду `chezmoi apply -v`. Программа снова выводит список изменений, с которыми пользователь теперь уже согласился.

```
+
+## status left/right sections separators
+tmux_conf_theme_left_separator_main=""
+tmux_conf_theme_left_separator_sub="|"
+tmux_conf_theme_right_separator_main=""
+tmux_conf_theme_right_separator_sub="|"
+##tmux_conf_theme_left_separator_main='\uE0B0' # /!\ you don't need to install Powerli
no
+##tmux_conf_theme_left_separator_sub='\uE0B1' # you only need fonts patched with
+##tmux_conf_theme_right_separator_main='\uE0B2' # Powerline symbols or the standalone
+##tmux_conf_theme_right_separator_sub='\uE0B3' # PowerlineSymbols.otf font, see READ
ME.md
+
+## status left/right content:
+## - separate main sections with "|"
+## - separate subsections with ","
+## - built-in variables are:
+## - ##{battery_bar}
+## - ##{battery_hbar}
+## - ##{battery_percentage} come, the more you are able to hear"
+## - ##{battery_status}
+## - ##{battery_vbar}
+## - ##{circled_session_name}
+## - ##{hostname_ssh}
+## - ##{hostname}
+## - ##{hostname_full}
```

Рис. 2.14: Часть вывода изменений

2.4 Ежедневные операции с chezmoi

С помощью `chezmoi` можно извлекать изменения из репозитория одной командой и, объединив команды, провести одновременно проверку изменений и сравнение с текущей версией. Вывод `'already up to date'` говорит о том, что данные в репозитории и клоне на устройстве актуальны друг относительно друга.

```
(tabuller@jordi)-[~]  
$ ./bin/chezmoi update  
Already up to date.  
  
(tabuller@jordi)-[~]  
$ ./bin/chezmoi git pull -- --autostash --rebase &6 chezmoi diff  
Already up to date.  
chezmoi: command not found  
  
(tabuller@jordi)-[~]  
$ ./bin/chezmoi git pull -- --autostash --rebase &6 ./bin/chezmoi diff  
Already up to date.
```

Рис. 2.15: Простые операции с chezmoi

Кроме этого, есть возможность настроить автоматическую отправку коммитов. Для этого необходимо изменить файл конфигурации (по умолчанию автофиксация и отправка изменений отключены):

```
GNU nano 7.2 /home/buller/  
[data]  
  email = "1132231835@pfur.ru"  
  
[git]  
  autoCommit = true  
  autoPush = true
```

Рис. 2.16: Редактирование файла конфигурации

3 Выводы

Получены навыки работы с менеджером паролей pass и использования chezmoi для управления файлами конфигурации домашнего каталога пользователя