

# Лабораторная работа № 5

## Цель работы

Создание ключей безопасности

## Выполнение лабораторной работы

### Менеджер паролей pass

Установка

pass

```
dnf install pass pass-otp gopass
```

```
dnf install gopass
```

###Настройка ####Ключи GPG

Просмотр списка ключей:

```
gpg --list-secret-keys
```

 Если ключа нет, нужно создать новый:

```
gpg --full-generate-key
```

 Инициализация хранилища

Инициализируем хранилище:

```
pass init
```

 Синхронизация с git

Создадим структуру git:

```
pass git init
```

 Также можно задать адрес репозитория на хостинге (репозиторий необходимо предварительно создать):

```
pass git remote add origin git@github.com:<git_username>/<git_repo>.git
```

 Для синхронизации выполняется следующая команда:

```
pass git pull
```

```
pass git push
```

 Прямые изменения

Следует заметить, что отслеживаются только изменения, сделанные через сам gopass (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 можно 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 native messaging.

Плагин browserpass

Репозиторий: <https://github.com/browserpass/browserpass-extension> (<https://github.com/browserpass/browserpass-extension>). Плагин для браузера Плагин для Firefox: <https://addons.mozilla.org/en-US/firefox/addon/browserpass-ce/> (<https://addons.mozilla.org/en-US/firefox/addon/browserpass-ce/>). Плагин для Chrome/Chromium: <https://chrome.google.com/webstore/detail/browserpass-ce/naepdomgkenhinolocfifgehidddafch> (<https://chrome.google.com/webstore/detail/browserpass-ce/naepdomgkenhinolocfifgehidddafch>). Интерфейс для взаимодействия с браузером (native messaging)

Репозиторий: <https://github.com/browserpass/browserpass-native> (<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 для вашего хранилища паролей; 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 и скачивает необходимый файл:

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="yamadharmadotfiles-template" --private
```

Подключение репозитория к своей системе Инициализируйте chezmoi с вашим репозиторием dotfiles:

chezmoi init git@github.com: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/dotfiles Или через ssh:
```

chezmoi init git@github.com: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 содержимым файла, файлом в вашей рабочей копии и изменённым содержимым файла:

chezmoi merge file\_name При существующем каталоге chezmoi можно получить и применить последние изменения из вашего репозитория:

```
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/dotfiles Через ssh:
```

```
chezmoi init --apply git@github.com:/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 затем `chezmoi apply`. Извлеките последние изменения из своего репозитория и посмотрите, что изменится, фактически не применяя изменения

Выполните:

`chezmoi git pull -- --autostash --rebase && chezmoi diff` Это запускается `git pull --autostash --rebase` в вашем исходном каталоге, а `chezmoi diff` затем показывает разницу между целевым состоянием, вычисленным из вашего исходного каталога, и фактическим состоянием. 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` следующее:

`[git] autoCommit = true autoPush = true` Всякий раз, когда в исходный каталог вносятся изменения, chezmoi фиксирует изменения с помощью автоматически сгенерированного сообщения фиксации и отправляет их в ваш репозиторий. Будьте осторожны при использовании `autoPush`. Если ваш репозиторий dotfiles является общедоступным, и вы случайно добавили секрет в виде обычного текста, этот секрет будет отправлен в ваш общедоступный репозиторий.