

Шаблон отчёта по лабораторной работе №5

Настройка рабочей среды.

Баптишта Матеуж Андре

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Установка и Настройка Pass	6
3.2	Настройка интерфейса с браузером	7
3.3	Сохранение пароля	7
3.4	Управление файлами конфигурации	8
3.4.1	Дополнительное программное обеспечение	8
3.5	Установка бинарного файла	9
3.6	Создание собственного репозитория	9
3.7	Подключение репозитория к своей системе	9
3.8	Использование chezmoi на нескольких машинах	10
3.9	Ежедневные операции с chezmoi	10
4	Выводы	12
	Список литературы	13

Список иллюстраций

3.1	Установка pass pass-otp	6
3.2	Установка gorpass	6
3.3	Создание ключа GPG	6
3.4	Созданные ключи	6
3.5	Инициализирование хранилище	6
3.6	посмотр статуса синхронизации	7
3.7	Добавлен плагин	7
3.8	Включение репозитория Copr	7
3.9	Установка browserpass native	7
3.10	Добавление пароля	7
3.11	Подтверждение созданного пароля	8
3.12	Замена пароля	8
3.13	Установка дополнительного программного обеспечения	8
3.14	Включение copr peterwu/iosevka	8
3.15	Поиск iosevka	8
3.16	Установка шрифты	8
3.17	Установка chezmoi	9
3.18	Создание репозиторий	9
3.19	Инициализирование Chezmoi	9
3.20	Проверка изменений	9
3.21	Применение изменений	9
3.22	Установка chezmoi на другой машине	10
3.23	Инициализирование Chezmoi на другой машине	10
3.24	Проверка изменений на второй машине	10
3.25	Применение изменений на второй машине	10
3.26	Обновление chezmoi	10
3.27	Обновление chezmoi	10
3.28	Извлечение изменений	11
3.29	Применение изменений после обновления	11
3.30	Функция фиксирование изменений автоматически	11

1 Цель работы

Цель данной работы – настройка рабочей среды с помощью менеджера паролей pass и конфигурации chezmoi.

2 Задание

1. Установить и настроить Pass
2. Настроить интерфейс с браузером
3. Управление файлами конфигурации
4. Использовать chezmoi на нескольких машинах

3 Выполнение лабораторной работы

3.1 Установка и Настройка Pass

С помощью командой `dnf install` через суперпользователя устанавливаю `pass`, `pass-otp` и `gopass`:

Установка `pass pass-otp`

Рис. 3.1: Установка `pass pass-otp`

Установка `gopass`

Рис. 3.2: Установка `gopass`

Создаю новый ключ GPG поскольку команда `gpg --list-secret-keys` ничего не выводила:

Создание ключа GPG

Рис. 3.3: Создание ключа GPG

Созданные ключи

Рис. 3.4: Созданные ключи

С помощью `pass init`, инициализирую хранилище указывая свой адрес электронной почты и создаю структуру `git` (`pass git init`) для синхронизации:

Инициализирование хранилище

Рис. 3.5: Инициализирование хранилище

Перехожу в ~/.password-store для просмотра статуса синхронизации:

посмотр статуса синхронизации

Рис. 3.6: просмотр статуса синхронизации

3.2 Настройка интерфейса с браузером

Добавляю плагин browserpass для firefox через браузера:

Добавлен плагин

Рис. 3.7: Добавлен плагин

Для взаимодействия с браузером используется интерфейс native messaging. Поэтому кроме плагина к браузеру устанавливается программа, обеспечивающая интерфейс native messaging. Делаю это через терминала

Включение репозитория Copr

Рис. 3.8: Включение репозитория Copr

Установка browserpass native

Рис. 3.9: Установка browserpass native

3.3 Сохранение пароля

Добавляю новый пароль в файл, который будет использоваться для хранения пароля. Этот файл находится в каталоге, определяющее файловую структуру для вашего хранилища паролей.

Добавление пароля

Рис. 3.10: Добавление пароля

При запуске pass BrowserPass/Passwords.txt, выводится запрос подтвердить созданный пароль:

Подтверждение созданного пароля

Рис. 3.11: Подтверждение созданного пароля

Заменяю существующий пароль с помощью `pass generate -in-place Passwords.txt`

Замена пароля

Рис. 3.12: Замена пароля

3.4 Управление файлами конфигурации

3.4.1 Дополнительное программное обеспечение

Устанавливаю дополнительное программное обеспечение:

Установка дополнительного программного обеспечения

Рис. 3.13: Установка дополнительного программного обеспечения

Устанавливаю шрифты `iosevka`. Для этого надо включить `copr peterwu/iosevka` и искать `iosevka`:

Включение `copr peterwu/iosevka`

Рис. 3.14: Включение `copr peterwu/iosevka`

Поиск `iosevka`

Рис. 3.15: Поиск `iosevka`

Установка шрифты

Рис. 3.16: Установка шрифты

3.5 Установка бинарного файла

Скрипт определяет архитектуру процессора и операционную систему и скачивает необходимый файл, поэтому я просто запускаю `sh -c "$(wget -qO- chezmoi.io/get)"`:

Установка chezmoi

Рис. 3.17: Установка chezmoi

3.6 Создание собственного репозитория

С помощью утилиты `gh` я создаю новый репозиторий `dotfiles` по шаблону `yamadharma`

Создание репозиторий

Рис. 3.18: Создание репозиторий

3.7 Подключение репозитория к своей системе

Chezmoi является клоном репозитория `dotfiles`. Инициализирую его:

Инициализирование Chezmoi

Рис. 3.19: Инициализирование Chezmoi

Проверяю какие изменения внесёт `chezmoi` в домашний каталог, запуская `chezmoi diff`:

Проверка изменений

Рис. 3.20: Проверка изменений

Применяю изменения запуская `chezmoi apply -v`:

Применение изменений

Рис. 3.21: Применение изменений

3.8 Использование chezmoi на нескольких машинах

На другой машине я устанавливаю chezmoi

Установка chezmoi на другой машине

Рис. 3.22: Установка chezmoi на другой машине

Далее инициализирую chezmoi со своим репозиторием dotfiles:

Инициализирование Chezmoi на другой машине

Рис. 3.23: Инициализирование Chezmoi на другой машине

Проверяю какие изменения внесёт chezmoi в домашний каталог, запуская chezmoi diff:

Проверка изменений на второй машине

Рис. 3.24: Проверка изменений на второй машине

Применяю изменения запуская chezmoi apply -v:

Применение изменений на второй машине

Рис. 3.25: Применение изменений на второй машине

Меня устраивают изменения, внесённые chezmoi, поэтому я просто обновляю chezmoi :

Обновление chezmoi

Рис. 3.26: Обновление chezmoi

3.9 Ежедневные операции с chezmoi

На моей основной машине я обновляю chezmoi на всякий случай:

Обновление chezmoi

Рис. 3.27: Обновление chezmoi

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

Извлечение изменений

Рис. 3.28: Извлечение изменений

Далее применяю “изменения”:

Применение изменений после обновления

Рис. 3.29: Применение изменений после обновления

Когда в исходный каталог вносятся изменения, `chezmoi` фиксирует изменения с помощью автоматически сгенерированного сообщения фиксации и отправляет их в репозиторий. Эта функция отключена по умолчанию но у меня уже была включена. Можно это увидеть в файле конфигурации `~/.config/chezmoi/chezmoi.toml`:

Функция фиксирование изменений автоматически

Рис. 3.30: Функция фиксирование изменений автоматически

4 Выводы

При выполнении данной работы я настроила рабочую среду с помощью менеджера паролей pass и конфигурации chezmoi.

Список литературы

Архитектура ЭВМ