МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И ЗАНЯТОСТИ НАСЕЛЕНИЯ

КРАЕВОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ПРОФЕССОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«НАХОДКИНСКИЙ ГОСУДАРСТВЕННЫЙ

ГУМАНИТАРНО-ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Дисциплина: Инженерно-технические средства защиты информации

Специальность: 100205 Обеспечения информационной безопасности автоматизированных систем

**Отчет**

Основы Python для информационной безопасности автоматизированных систем

|  |  |  |
| --- | --- | --- |
| Студент группы 632 очной формы обучения |  | Бондаренко П.Е. |
|  |  |  |
|  |  |  |
| Руководитель |  |  |
|  |  |  |
|  |  |  |
| Нормоконтролер |  |  |

г. Находка, 2023

**Содержание**

[1. Основы Python для информационной безопасности автоматизированных систем 3](#_Toc149587999)

[2. Освоение базовых команд Python 3](#_Toc149588000)

[3. Разработка скрипта для проверки силы пароля 3](#_Toc149588001)

[4. Анализ исходных кодов на наличие уязвимостей 4](#_Toc149588002)

[Рекомендация по улучшению 5](#_Toc149588003)

[Вывод 5](#_Toc149588004)

1. Основы Python для информационной безопасности автоматизированных систем

**Цель:** Познакомиться с основами Python в контексте информационной безопасности, а также разработать простой инструмент для анализа безопасности.

**Задачи:**

1) Освоить базовые команды Python

2) Разработать скрипт для проверки силы пароля

3) Анализировать исходные коды на наличие уязвимостей

2. Освоение базовых команд Python

Написать программу, которая запросит имя пользователя и в конце поприветствует.

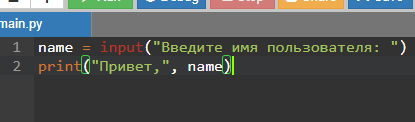


Рис. 1 – Ввод имени и приветствие

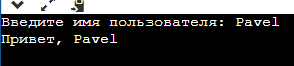


Рис. 2 – Консоль

3. Разработка скрипта для проверки силы пароля

Написать программу, которая будет принимать пароль на вход и анализировать его на соответствие данным требованиям. Программа должна выводить результат анализа.

**Пароль считается надёжным, если он содержит:**

* Не менее 8 символов.
* Как минимум одну заглавную букву.
* Как минимум одну строчную букву.
* Как минимум одну цифру.
* Как минимум один специальный символ (например, !@#$%^&\*).

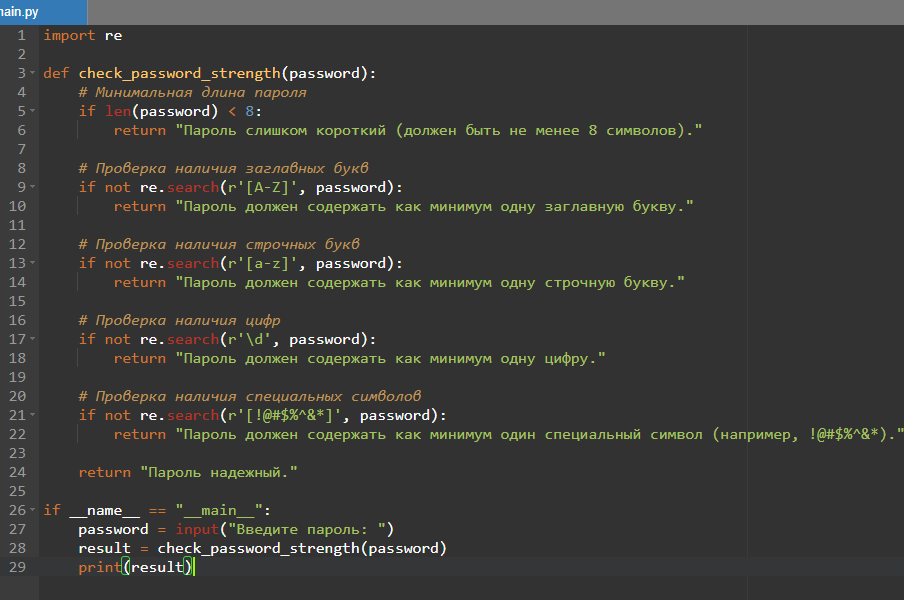


Рис. 3 – Проверка силы пароля

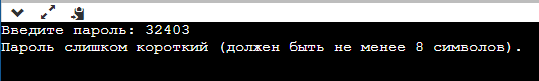


Рис. 4 – Не удовлетворяет

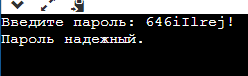


Рис. 5 – Пароль надежный

4. Анализ исходных кодов на наличие уязвимостей

def login(user, password):

stored\_password = "secret123"

if password == stored\_password:

print(f"Welcome, {user}!")

else: print("Incorrect password!")

Проанализировать данный код и найти уязвимости (написать рекомендацию по его улучшению)

Уязвимости

1) Хранение пароля в открытом виде: Пароль хранится в явном виде в коде (stored\_password = "secret123"), что является серьезной уязвимостью. Любой, кто имеет доступ к исходному коду, сможет узнать пароль.

2) Отсутствие хэширования пароля: Пароль не хэшируется перед сравнением. Без хэширования, пароль остается в текстовом виде в памяти и в коде, что делает его уязвимым для несанкционированного доступа.

3) Отсутствие защиты от перебора пароля: Этот код не включает в себя какие-либо механизмы защиты от атак перебора пароля (brute force), что оставляет систему уязвимой к таким атакам.

Рекомендация по улучшению

* Хэшировать пароль перед сохранением в базу данных;
* Добавить комментарии к коду;
* Использовать булево значение для возврата результата: Вместо прямого вывода на экран, возвращает булевое значение, которое показывает, успешен ли вход или нет;

Вывод

Python позволяют использовать для автоматизации задач, анализа данных, работы с сетями и создания инструментов для защиты информации. Python предоставляет множество библиотек и инструментов, которые делают его мощным инструментом в области информационной безопасности.