**Задача 3: Простая проверка лицензии**

1. Изучите простейший пример проверки файла лицензии:

import os

def check\_license():

if not os.path.exists("license.key"):

print("Файл лицензии не найден!")

return False

with open("license.key") as f:

key = f.read().strip()

return key == "ABCD-1234-EFGH"

def main():

if not check\_license():

print("Программа заблокирована!")

return

print("Программа запущена!")

*# Основной код*

if \_\_name\_\_ == "\_\_main\_\_":

Данный код, но с комментариями на каждой строке

import os # Импорт модуля для работы с операционной системой

def check\_license(): # Функция проверки лицензии

if not os.path.exists("license.key"): # Проверка наличия файла лицензии

print("Файл лицензии не найден!") # Сообщение об отсутствии файла

return False # Возврат False при отсутствии файла

with open("license.key") as f: # Открытие файла лицензии

key = f.read().strip() # Чтение и очистка ключа от пробелов

return key == "ABCD-1234-EFGH" # Проверка соответствия ключа

def main(): # Основная функция программы

if not check\_license(): # Проверка лицензии

print("Программа заблокирована!") # Сообщение о блокировке

return # Выход из программы

print("Программа запущена!") # Сообщение об успешном запуске

# Основной код программы

if \_\_name\_\_ == "\_\_main\_\_": # Проверка запуска как основного скрипта

main() # Вызов основной функции

Проверка программы на трех случаях:

Нет файла лицензии:

Вывод: "Файл лицензии не найден!"

Затем: "Программа заблокирована!"

Программа завершает работу

Есть файл лицензии с правильным ключом "ABCD-1234-EFGH":

Вывод: "Программа запущена!"

Программа продолжает выполнение основного кода

Лицензия не соответствует (любой другой ключ):

Вывод: "Программа заблокирована!"

Программа завершает работу

Задача 4: Ответы на вопросы

1. Защита интеллектуальной собственности, Предотвращение несанкционированного доступа, Сохранение конкурентных преимуществ и Защита пользовательских данных
2. Сохранение доходов от продажи ПО, Защита репутации компании, Предотвращение утечки конфиденциальных алгоритмов, Соответствие нормативным требованиям
3. Кража интеллектуальной собственности, Обход лицензионных ограничений, Внедрение вредоносного кода
4. Это анализ программы для понимания ее работы без доступа к исходному коду. Из угроз стоит отметить раскрытие алгоритмов, обход защиты, создание копий
5. Декомпиляция, Динамический анализ (отладка), Анализ сетевого трафика, Дамп памяти
6. Чтение незашифрованных строк, Обход проверок лицензии, Модификация исполняемого кода, Инъекция кода
7. Обфускация кода, Шифрование, Использование нативного кода, Лицензирование
8. Преобразование данных в нечитаемый вид с помощью ключа. Виды – симметричное, асимметричное, хеширование
9. Преобразование кода в трудночитаемую форму с сохранением функциональности. Помогает затруднить анализ и модификацию кода
10. PyArmor (Python), Obfuscator-LLVM (C/C++), ProGuard (Java), Dotfuscator (.NET)
11. Контроль использования ПО, Ограничение функциональности без лицензии, Привязка к оборудованию или пользователю
12. Аппаратные ключи, Онлайн-активация, Криптографически подписанные лицензии
13. Попытка декомпиляции, Анализ в отладчике, Проверка на изменение системного времени, Тесты на обход лицензирования
14. Своевременное выявление уязвимостей, Соответствие стандартам безопасности, Поддержание доверия клиентов
15. Регулярные проверки на уязвимости, Тестирование на различных платформах, Интеграция в CI/CD pipeline
16. Использование аппаратных ключей, Комбинирование обфускации и шифрования, Серверная проверка лицензий
17. Для мобильных приложений – проверка root/джейлбрейка

Для веб-приложений – защита API

Для десктопных – защита от модификации

1. Взлом защиты игр Denuvo, Обход лицензирования Adobe Photoshop, Кейсы с банковским ПО
2. Анализ поведения на предмет аномалий, Генерация уникальных защитных механизмов, Обнаружение попыток взлома
3. Законы об авторском праве
4. Использование блокчейна для лицензирования, Биометрическая аутентификация, Контейнеризация приложений
5. Усложнение методов анализа, Необходимость более сложной защиты, Развитие инструментов для обеих сторон
6. Атаки на устройства с ограниченными ресурсами, Уязвимости в моделях машинного обучения, Проблемы безопасности в распределенных системах