

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет безопасности информационных технологий**

**Дисциплина:**

*« Технологии и методы программирования »*

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

Выполнил:

Чу Ван Доан – Студент группы N3347



Преподаватель:

Ищенко Алексей Петрович

Санкт-Петербург

2024 г.

## СОДЕРЖАНИЕ

<b>Задания.....</b>	<b>3</b>
<b>ХОД РАБОТЫ.....</b>	<b>4</b>
1. Задание 1а.....	4
2. Задание 1б.....	7
<b>ВЫВОД.....</b>	<b>13</b>

## **Задания**

### **Задание 1(а): Локальная программа**

– Разработать программу, запрещающую в текущем (том, котором она находится) каталоге создание, копирование, удаление или переименование файлов с заданными именами (можно использовать маски файлов). Список имен или их шаблонов хранить в файле `template.tbl`, как текст. Должна быть обеспечена защита этого файла от удаления, несанкционированного просмотра и модификации. При установке программы можно предусмотреть ее отключение с использованием пароля, хранящегося в первой строке файла `template.tbl` в хешированном виде. Программа должна включать и выключать режим защиты.

– Для ОС Майкрософт рекомендуется (хотя может и устарели) использовать: `SetFileSecurity` и `SetNamedSecurityInfo`, а также другие материалы из источников в И-нете:

<http://msdn.microsoft.com/en-us/library/windows/desktop/aa379577%28v=vs.85%29.aspx>

– Возможна реализация программы и под Линукс.

### **Задание 1(б): Веб-скрипт**

– Реализовать скрипт, внедряемый в код HTML-документа, который реализует защиту от копирования в буфер (нет возможности выделять содержимое, копировать в буфер как текст или через скриншот и т.д.) и сохранения всех страниц, вызываемых с текущей. При этом печать этих страниц на бумажный носитель должна быть доступна. Отключение скрипта должно происходить с использованием пароля, хранящегося в теле скрипта в зашифрованном виде.

## ХОД РАБОТЫ

### 1. Задание 1а

#### 1.1 Код: <Python>

```
import os
import hashlib
import fnmatch
import stat
import subprocess

TEMPLATE_FILE = 'template.tbl'

# функция для чтения файла template.tbl
def read_template_file(filename):
    with open(filename, 'r') as file:
        lines = file.readlines()
        hashed_password = lines[0].strip() # Первая строка содержит
        # хешированный пароль
        forbidden_files = [line.strip() for line in lines[1:]] # Список
        # запрещённых файлов
    return hashed_password, forbidden_files

# функция для хеширования пароля
def hash_password(password):
    return hashlib.sha256(password.encode()).hexdigest()

# функция проверки пароля пользователя
def check_password(hashed_password, input_password):
    return hashed_password == hash_password(input_password)

# функция проверки, является ли файл запрещённым
def is_forbidden_file(filename, forbidden_files):
    for pattern in forbidden_files:
        if fnmatch.fnmatch(filename, pattern):
            return True
    return False

# функция для защиты файла (запрет на чтение, запись, удаление,
# перемещение)
def protect_file(filepath):
```

```

# Установка прав без чтения, записи и выполнения для всех
пользователей
os.chmod(filepath, 0)
# Используем chattr для предотвращения удаления и перемещения файла
(immutable)
subprocess.run(['sudo', 'chattr', '+i', filepath])
print(f"Файл защищён: {filepath}")

# Функция для снятия защиты с файла (разрешение чтения, записи,
удаления, перемещения)
def unprotect_file(filepath):
    # Снять атрибут immutable (разрешить удаление файла)
    subprocess.run(['sudo', 'chattr', '-i', filepath])
    # Восстановить права на чтение, запись и выполнение для владельца
    os.chmod(filepath, stat.S_IRWXU)
    print(f"Доступ к файлу восстановлен: {filepath}")

# Функция для защиты или снятия защиты файлов в текущем каталоге
def protect_files(directory, forbidden_files, enable=True):
    for root, dirs, files in os.walk(directory):
        for file in files:
            if is_forbidden_file(file, forbidden_files):
                filepath = os.path.join(root, file)
                if enable:
                    protect_file(filepath)
                else:
                    unprotect_file(filepath)

# Функция для включения/выключения режима защиты
def toggle_protection(template_file, enable=True):
    hashed_password, forbidden_files = read_template_file(template_file)

    input_password = input("Введите пароль для включения/выключения
защиты: ")

    if check_password(hashed_password, input_password):
        if enable:
            print("Пароль правильный! Начинаем защищать файлы.")
        else:
            print("Пароль правильный! Снимаем защиту с файлов.")
        current_directory = os.getcwd() # Текущая директория
        protect_files(current_directory, forbidden_files, enable)
    else:

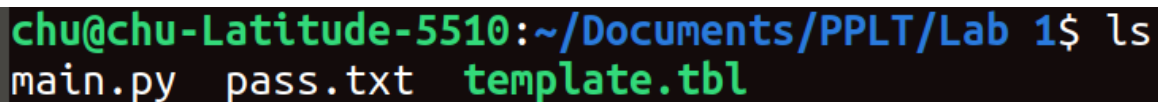
```

```

        print("Неправильный пароль! Не могу включить/выключить защиту.")

# Запуск программы
if __name__ == "__main__":
    choice = input("Введите 'on' для включения защиты или 'off' для
отключения защиты: ").strip().lower()
    if choice == 'on':
        toggle_protection(TEMPLATE_FILE, enable=True)
    elif choice == 'off':
        toggle_protection(TEMPLATE_FILE, enable=False)
    else:
        print("Неверный выбор.")

```



```

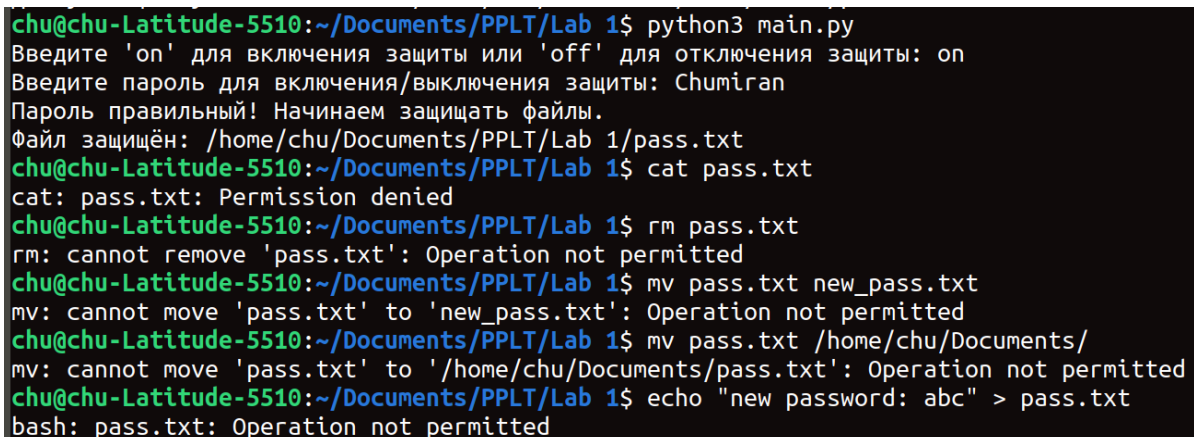
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ ls
main.py  pass.txt  template.tbl

```

Рисунок 1 – Файлы в каталоге

- lab1.py — основная программа.
- template.tbl — это файл, в котором хранятся имена файлов, которые необходимо защитить. Первая строка — это зашифрованный пароль, используемый для аутентификации при включении или выключении уровня защиты.
- pass.txt — это файл, который необходимо защитить.

## 1.2 Запустить программу



```

chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ python3 main.py
Введите 'on' для включения защиты или 'off' для отключения защиты: on
Введите пароль для включения/выключения защиты: Chumigan
Пароль правильный! Начинаем защищать файлы.
Файл защищён: /home/chu/Documents/PPLT/Lab 1/pass.txt
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ cat pass.txt
cat: pass.txt: Permission denied
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ rm pass.txt
rm: cannot remove 'pass.txt': Operation not permitted
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ mv pass.txt new_pass.txt
mv: cannot move 'pass.txt' to 'new_pass.txt': Operation not permitted
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ mv pass.txt /home/chu/Documents/
mv: cannot move 'pass.txt' to '/home/chu/Documents/pass.txt': Operation not permitted
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ echo "new password: abc" > pass.txt
bash: pass.txt: Operation not permitted

```

Рисунок 2. Рисунок 2 – Включение режима защиты файлов

```

chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ python3 main.py
Введите 'on' для включения защиты или 'off' для отключения защиты: off
Введите пароль для включения/выключения защиты: Chumiran
Пароль правильный! Снимаем защиту с файлов.
Доступ к файлу восстановлен: /home/chu/Documents/PPLT/Lab 1/pass.txt
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ cat pass.txt
My password is: Chumiran
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ echo "New password is abc" > pass.txt
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ cat pass.txt
New password is abc
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ mv pass.txt new_pass.txt
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ ls
main.py  new_pass.txt  template.tbl
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ rm new_pass.txt
chu@chu-Latitude-5510:~/Documents/PPLT/Lab 1$ ls
main.py  template.tbl

```

Рисунок 3 – Выключение режима защиты файлов

## 2. Задание 16

### 2.1 Код: <HTML>

```

<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Защита контента</title>

  <style>
    /* Центрирование всего содержимого */
    body, html {
      height: 100%;
      margin: 0;
      font-family: Arial, sans-serif;
      display: flex;
      justify-content: center; /* Центрирование по горизонтали */
      align-items: center;     /* Центрирование по вертикали */
      background-size: cover;
      background-position: center;
    }

    /* Центрирование блока контента */
    #main-content {
      background-color: rgba(255, 255, 255, 0.8); /* Полупрозрачный
белый фон для удобства чтения */
      padding: 20px;
      border-radius: 10px;
    }
  </style>

```

```

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
    text-align: center;
}

/* CSS для отключения выделения текста и правого клика при
включенной защите */
.protected-content {
    -webkit-user-select: none; /* Safari */
    -moz-user-select: none; /* Firefox */
    -ms-user-select: none; /* Internet Explorer/Edge */
    user-select: none; /* Общая версия */
    pointer-events: none; /* Отключение событий мыши */
}

/* CSS для отображения опций включения/выключения защиты */
#protection-options {
    margin: 20px;
}

#password-section {
    margin-top: 10px;
}

button {
    padding: 10px 20px;
    margin: 5px;
    font-size: 16px;
    cursor: pointer;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 5px;
    transition: background-color 0.3s;
}

button:hover {
    background-color: #45a049;
}

input[type="password"] {
    padding: 10px;
    font-size: 16px;
    margin-right: 10px;
}

```



```

        border: 1px solid #ccc;
        border-radius: 5px;
    }
</style>
</head>
<body>
    <!-- Основной блок контента -->
    <div id="main-content">

        <!-- Защищённый контент -->
        <div id="content" class="protected-content">
            <h1>Защита от копирования</h1>
            <p>Этот контент защищен от копирования и скриншотов, если режим
защиты включен.</p>
        </div>

        <!-- Блок опций включения/выключения защиты -->
        <div id="protection-options">
            <button onclick="promptPassword('enable') ">Включить
защиту</button>
            <button onclick="promptPassword('disable') ">Отключить
защиту</button>
        </div>

        <!-- Форма для ввода пароля -->
        <div id="password-section" style="display: none;">
            <input type="password" id="password" placeholder="Введите пароль">
            <button onclick="handlePassword() ">Подтвердить</button>
        </div>
    </div>

    <script>
        // SHA256 хеш пароля
        const hashedPassword =
"9f836f77a5918d1d48f774cbde22f4317f7931018ccd44e4350a441e00c6b56a";

        // Переменная для состояния действия включения/выключения защиты
        let protectionAction = "";

        // Показать форму для ввода пароля в зависимости от действия
        function promptPassword(action) {
            protectionAction = action;

```

```

        document.getElementById("password-section").style.display =
"block";
    }

    // Обработка введённого пароля
    async function handlePassword() {
        const inputPassword = document.getElementById("password").value;

        // Хеш SHA256 введённого пароля
        const inputHashed = await sha256(inputPassword);

        if (inputHashed === hashedPassword) {
            if (protectionAction === 'enable') {
                enableProtection();
                alert("Защита включена. Вы можете только распечатать страницу
и отключить защиту.");
            } else if (protectionAction === 'disable') {
                disableProtection();
                alert("Защита отключена. Вы можете взаимодействовать с
контентом.");
            }
            document.getElementById("password-section").style.display =
"none"; // Скрыть форму после подтверждения
        } else {
            alert("Неверный пароль!");
        }
    }

    // Включить защиту
    function enableProtection() {
        // Отключить взаимодействие с контентом, кроме кнопок защиты

        document.getElementById("content").classList.add("protected-content");
    }

    // Отключить защиту
    function disableProtection() {
        // Включить взаимодействие с контентом

        document.getElementById("content").classList.remove("protected-content"
);
    }
}

```

```

// Хеширование SHA256 (с использованием SubtleCrypto API браузера)
async function sha256(message) {
  const msgBuffer = new TextEncoder().encode(message);
  const hashBuffer = await crypto.subtle.digest('SHA-256',
msgBuffer);
  const hashArray = Array.from(new Uint8Array(hashBuffer));
  return hashArray.map(b => b.toString(16).padStart(2,
'0')).join('');
}
</script>
</body>
</html>

```



127.0.0.1:5500 says  
 Защита отключена. Вы можете взаимодействовать с  
 контентом.

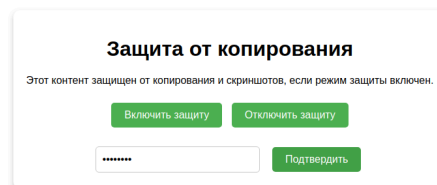
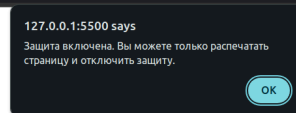
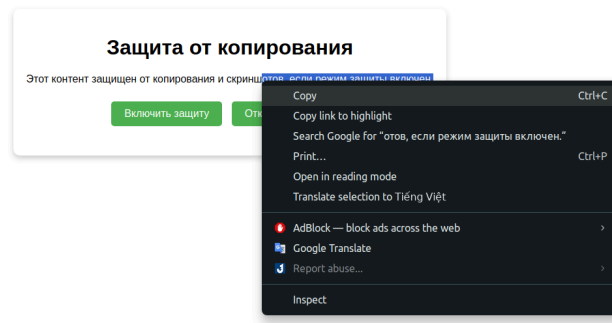
OK

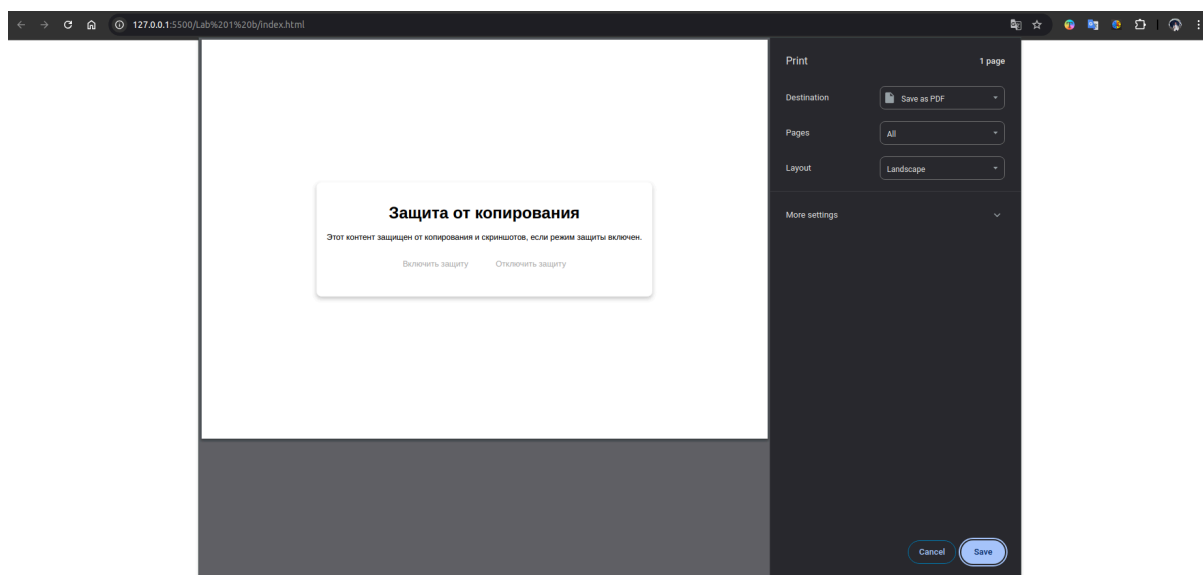
**Защита от копирования**

Этот контент защищен от копирования и скриншотов, если режим защиты включен.

Включить защиту Отключить защиту

\*\*\*\*\* Подтвердить





## ВЫВОД

В лаборатории мы подошли к методам программирования для защиты файлов и веб-сайтов. Использовала язык Python, чтобы предотвратить удаление, редактирование, переименование файлов и т. д. И манипулировала языком HTML, чтобы предотвратить хранение и копирование контента с веб-сайтов. Это обеспечивает лучшую безопасность контента при программировании.