Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

**ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ**

Отчет по выполнению лабораторной работы №1

Выполнил

ст. гр. 230781 А. Р. Ивлев

Проверил

Асс. Греков Михаил Михайлович

Тула 2022

**Лабораторная работа №1**

**Защита от разрушающих программных воздействий, защита программ от изменения и контроль целостности**

**Цель работы**

Познакомиться с общими принципами защиты программного обеспечения и способами организации контроля целостности исполняемых модулей и важных

программных данных.

**Задание на работу**

Вариант №4

Разработать программу по контролю целостности определенных директорий.

**Ход работы**

На языке Python был написан программа, которая вычисляет MD5-хэш для каждого файла в новой директории и позволяет проверить все файлы в ранее просчитанной директории.

*Листинг 1. Код программы*

import hashlib

from tkinter import filedialog

from tkinter import \*

from tkinter import ttk

import os

from termcolor import colored

hashes = dict()

# Загружает ранее сохраненные рассчитанные хэши файлов

def load():

filename = filedialog.askopenfilename()

with open(filename, 'r') as db:

text = db.read().split('\n')

for string in text:

if string == '':

continue

file, hash = string.split('\t')

hashes[file] = hash

# Проверяет директорию на наличие измененных файлов

def check\_directory():

global hashes

root = filedialog.askdirectory()

if root == '':

return

for subdir, \_, files in os.walk(root):

for name in files:

file = os.path.join(subdir, name).replace('\\', '/')

print(file + ":\t", end='')

text = open(file, 'rb').read()

hash = hashlib.md5(text).hexdigest()

try:

if hashes[file] == hash:

print(colored('OK', 'green'))

else:

print(colored('Failed', 'red'))

except:

print(colored('Not Found', 'yellow'))

# Вычисляет хэши каждого файла в директории

def calc\_directory():

global hashes

root = filedialog.askdirectory()

if root == '':

return

for subdir, \_, files in os.walk(root):

for name in files:

file = os.path.join(subdir, name).replace('\\', '/')

text = open(file, 'rb').read()

hash = hashlib.md5(text).hexdigest()

hashes[file] = hash

print(colored('END', 'green'))

# Сохраняет ранее вычисленные хэши в файл

def save():

filename = filedialog.asksaveasfilename()

with open(filename, 'w') as db:

for file in hashes:

db.write(file + '\t' + (hashes[file]) + '\n')

# Окно выбора действий

root = Tk()

frm = ttk.Frame(root, padding=10)

frm.grid()

ttk.Button(frm, text="Load...", command=load).grid(column=0, row=0)

ttk.Button(frm, text="Calc...", command=calc\_directory).grid(column=0, row=1)

ttk.Button(frm, text="Check...", command=check\_directory).grid(column=0, row=2)

ttk.Button(frm, text="Save...", command=save).grid(column=0, row=3)

root.mainloop()

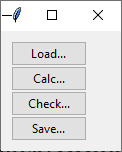


Рисунок 1. Окно программы

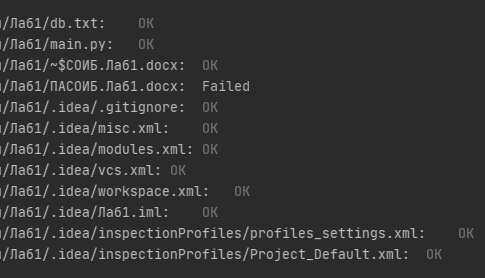


Рисунок 2. Результат работы программы

**Вывод**

Была разработана программа по контролю целостности определенных директорий.

Одними из возможных способов защиты ПО от подделки являются электронная цифровая подпись и контрольная сумма. Контрольная сумма – некоторое значение, рассчитанное по набору данных путѐм применения определенного алгоритма и используемое для проверки целостности данных при их передаче или хранении. Одним из самых популярных алгоритмов вычисления контрольной суммы файла является алгоритм MD5. Он является легко вычислимым и относительно сложно подделываемым, что может использоваться в простых задачах проверки файлов на неизменность.