|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| **Институт комплексной безопасности и специального приборостроения** | | |
|  |  |  |

Кафедра КБ-4 «Управление и моделирование систем»

**Отчет**

По проекту:  
Система шифрования ExFil

Выполнил:

Студент 3 курса

Группа ББСО-03-19

Шифр 19И1317

Кирнев Денис Сергеевич

Москва, 2021

Название проекта ExFil произошло от Random Access File Encryption System, Acc в слове Access схоже по звучанию с Ex, а Fil взято из слова File. В переводе это означает «Система шифрования файлов с произвольным доступом».

Опишем все алгоритмы и принцип работы программы в следующих пяти пунктах.

1. Интерфейс

С помощью интерфейса пользователь сможет взаимодействовать с программой данного проекта.

def interface(): #здесь находится интерфейс  
 print("Добро пожаловать в программу шифрования данных ExFil!")  
 print("Посмотреть список команд можно командой \"commands\"")  
 isRun = True  
 command = ""  
 while isRun:  
 command = input("\nx---> ")  
 print()  
 if command=="commands":

2. Список команд, информация и инструкция

При необходимости пользователь может ознакомиться со списком команд, информацией и инструкцией к программе:

if command=="commands":  
 print("\*-\*-\*-\*--->>> Команды ExFil <<<---\*-\*-\*-\*")  
 print("commands - пулучить список команд")  
 print("information - получить информацию о программе")  
 print("instruction - получить инструкцию к программе")  
 print("encrypt - зашифровать все файлы в указанной папке")  
 print("decrypt - расшифровать все файлы в указанной папке")  
 print("evaluate - выполненить однострочную команду в интерпретаторе Python")  
 print("system - выполненить однострочную команду в системной консоли")  
 print("setram - установить размер буфера")  
 print("getram - получить размер буфера")  
 print("setmin - установить минимальный размер файла-пароля")  
 print("getmin - получить минимальный размер файла-пароля")  
 print("setmax - установить максимальный размер файла-пароля")  
 print("getmax - получить максимальный размер файла-пароля")  
 print("setmainpath - установить главный путь")  
 print("getmainpath - получить главный путь")  
 print("isdir - проверка принадлежности указанного пути к папке")  
 print("isfile - проверка принадлежности указанного пути к файлу")  
 print("createfile - создать новый файл")  
 print("createdir - создать новую папку")  
 print("exists - проверка папки или файла на существование")  
 print("countfiles - посчитать, сколько файлов в указанной папке")  
 print("getfile - получить путь к файлу по его порядковому номеру в указанной папке (индексация от нуля)")  
 print("clear - очистить файл")  
 print("delete - удалить папку или файл")  
 print("size - определить размер папки или файла")  
 print("getabsolutepath - получить абсолютный путь к папке или файлу")  
 print("read - прочитать файл")  
 print("rewrite - перезаписать файл")  
 print("write - дозаписать в файл")  
 print("append - дозаписать в файл")  
 print("rename - переименовать папку или файл")  
 print("copy - скопировать папку или файл")  
 print("move - переместить папку или файл")  
 print("! - игнорировать следующее сообщение")  
 print("exit - выйти из программы")  
 print("stop - выйти из программы")  
 print("\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*")

3. Алгоритм создания файла-пароля

Перед шифрованием файлов в указанной папке, там должен находиться файл-пароль, с наименованием exfil.exfil, что соответствует названию проекта. Этот файл имеет случайный размер в диапазоне от minsize, до maxsize включительно. Все файлы, которые расположены в mainpath, сканируются и из случайно выбранного из них файла берётся кусок данных случайного размера со случайным сдвигом. Алгоритмы построены так, чтобы исключить большинство ошибок в процессе работы программы. К примеру, если из-за сдвига и/или считывания указатель достиг конца файла, то он перемещается в самое начало этого файла.

def createpasswordfile(path):  
 f = open(path+"\\exfil.exfil", "wb")  
 size = random.randint(filesio.minsize, filesio.maxsize)  
 arr = bytearray(size)  
 filescounter = filesio.countfiles(filesio.mainpath)  
 i = 0  
 c = 0  
 while i < size:  
 tempfile = " "  
 while tempfile==" ":  
 c = random.randint(0, filescounter)  
 if filesio.size(filesio.getFileByID(filesio.mainpath, c))>0:  
 tempfile = filesio.getFileByID(filesio.mainpath, c)  
 tf = open(tempfile, "rb")  
 tf.seek(random.randint(0, filesio.size(tempfile)-1))  
 length = random.randint(1, size-i)  
 c = 0  
 while c < length:  
 if tf.tell() == filesio.size(tempfile):  
 tf.seek(0)  
 t = bytearray(tf.read(1))  
 arr[i+c] = t[0]  
 print(str(i+c+1) + " of " + str(size))  
 c += 1  
 i+=length  
 f.write(arr)  
 f.flush()  
 f.close()

4. Алгоритм шифрования

К каждому байту каждого файла в указанной папке прибавляется значение байта из файла-пароля exfil.exfil со сдвигом в размер файла делённого по модулю на размер файла-пароля. В случае достижения указателя конца файла-пароля, указатель перемещается в начало этого файла-пароля.

def encrypt(path):  
 print("Создание файла-пароля...")  
 filesio.createpasswordfile(path)  
 print("Файл-пароль exfil.exfil успешно создан!")  
 counter = 0  
 filecounter = filesio.countfiles(path)  
 passwordfile = open(path+"\\exfil.exfil", "rb")  
 passwordfilesize = filesio.size(path+"\\exfil.exfil")  
 while counter < filecounter:  
 if filecounter-counter-1 > 0:  
 print("Осталось зашифровать еще: " + str(filecounter-counter-1) + " файлов!")  
 filesize = filesio.size(filesio.getFileByID(path, counter))  
 currentfilepath = filesio.getFileByID(path, counter)  
 currentfile = open(currentfilepath, "r+b")  
 seekpos = 0  
 tellpos = 0  
 if filesize>0 and currentfilepath!=path+"\\exfil.exfil":  
 passwordfile.seek(filesize%passwordfilesize)  
 tempsize = filesize  
 while tempsize>0:  
 if tempsize>=filesio.MEMORY:  
 data1 = currentfile.read(filesio.MEMORY)  
 tempsize-=filesio.MEMORY  
 else:  
 data1 = currentfile.read(tempsize)  
 tempsize-=tempsize  
 arr = bytearray(data1)  
 i = 0  
 if passwordfilesize>0:  
 while i < len(arr):  
 if (passwordfile.tell() == passwordfilesize):  
 passwordfile.seek(0)  
 t = passwordfile.read(1)  
 arr[i] = (arr[i]+t[0])%256  
 i+=1  
 tellpos = currentfile.tell()  
 currentfile.seek(seekpos)  
 currentfile.write(arr)  
 currentfile.seek(tellpos)  
 seekpos = tellpos  
 currentfile.flush()  
 currentfile.close()  
 counter+=1

5. Алгоритм дешифрования

Алгоритм дешифрования имеет обратный алгоритму шифрования принцип действия. Все значения, которые прибавляются в алгоритме шифрования — тут отнимаются. А также, в конце работы алгоритма дешифрования происходит удаление файла-пароля exfil.exfil.

def decrypt(path):  
 counter = 0  
 filecounter = filesio.countfiles(path)  
 passwordfile = open(path+"\\exfil.exfil", "rb")  
 passwordfilesize = filesio.size(path+"\\exfil.exfil")  
 while counter < filecounter:  
 if filecounter-counter-1 > 0:  
 print("Осталось расшифровать еще: " + str(filecounter-counter-1) + " файлов!")  
 filesize = filesio.size(filesio.getFileByID(path, counter))  
 currentfilepath = filesio.getFileByID(path, counter)  
 currentfile = open(currentfilepath, "r+b")  
 seekpos = 0  
 tellpos = 0  
 if filesize>0 and currentfilepath!=path+"\\exfil.exfil":  
 passwordfile.seek(filesize%passwordfilesize)  
 tempsize = filesize  
 while tempsize>0:  
 if tempsize>=filesio.MEMORY:  
 data1 = currentfile.read(filesio.MEMORY)  
 tempsize-=filesio.MEMORY  
 else:  
 data1 = currentfile.read(tempsize)  
 tempsize-=tempsize  
 arr = bytearray(data1)  
 i = 0  
 if passwordfilesize>0:  
 while i < len(arr):  
 if (passwordfile.tell() == passwordfilesize):  
 passwordfile.seek(0)  
 t = passwordfile.read(1)  
 arr[i] = (arr[i]-t[0])%256  
 i+=1  
 tellpos = currentfile.tell()  
 currentfile.seek(seekpos)  
 currentfile.write(arr)  
 currentfile.seek(tellpos)  
 seekpos = tellpos  
 counter+=1  
 passwordfile.close()  
 currentfile.flush()  
 currentfile.close()  
 filesio.delete(path+"\\exfil.exfil")

Все алгоритмы и принцип действия работы программы перечислены. Также перечислены только самые актуальные части кода. Остальные части кода тоже немаловажны, поскольку они создают дополнительную функциональность программы, однако они не перечисляют основные принципы действия алгоритмов и работы программы, поэтому тут они не указаны.