Федеральное государственное бюджетное образовательное учреждение высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации».

## Кафедра «Прикладная Информатика»

ОТЧЕТ

О ПРОДЕЛАННОЙ ЛАБОРАТОРНОЙ РАБОТЕ №4

по курсу «Информационная безопасность»

Выполнил: студент группы

ИК-731

**(наименование группы)**

***Соколов Дмитрий Александрович***

**(Ф.И.О.)**

## Нижний Новгород

**2025 г.**

ЛАБОРАТОРНАЯ РАБОТА № 4

«Защита от копирования. Привязка к аппаратному обеспечению. Использование реестра»

Цель работы: ознакомиться с возможностями «привязки» к характеристикам компьютера.

Вариант № 18

**Задание**

Разработать программу, реализующую привязку к компьютеру, используя совокупность характеристик согласно варианту задания. Добиться того, чтобы программа не запускалась на другом компьютере.

Привязка по MAC-адрес сетевой карты, тактовая частота процессора в соответствии с вариантом.

**Теория для создания приложения**

Цель привязки программы к компьютеру - необходимо запретить запуск программы на других пк, взяв за эталонные значения mac сетевой карты и тактовую частоту процессора моего пк. При не совпадении значений – запрет.

**Характеристики компьютера**

В качестве уникальных признаков я использую:

MAC-адрес сетевой карты - уникальный физический идентификатор сетевого адаптера и тактовую частоту cpu.

**Создание идентификатора ПК**

Идентификатор машины в моем случае это MAC + CPU speed. Например, объединение MAC-адреса и тактовой частоты процессора создаёт уникальную строку, которая практически никогда не повторяется на другом компьютере. Этот идентификатор используется как ключ для проверки запуска программы.

**Как работает?**

Программа хранит в отдельном файле зашифрованный идентификатор ПК. При запуске приложение заново собирает текущие характеристики и формирует новый идентификатор. Если новый идентификатор совпадает с сохранённым, запуск разрешён; если нет-программа блокируется.

**Как я обеспечу безопасность идентификатора?**

Я воспользовалсяхеш-функцией SHA256, поэтому в файле лицензии хранится только хеш, без раскрытия исходных данных. Таким образом, даже если злоумышленник получит файл лицензии, он не сможет легко восстановить MAC и частоту процессора для обхода проверки.

**Листинг программы и описание функций**

Для создания идентификатора ПК сначала предоставляю реальные характеристики.

MAC-сетевой карты:

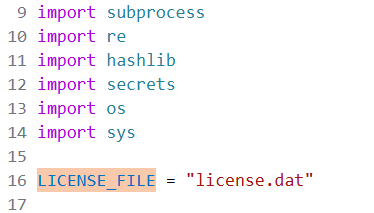


Частота процессора:

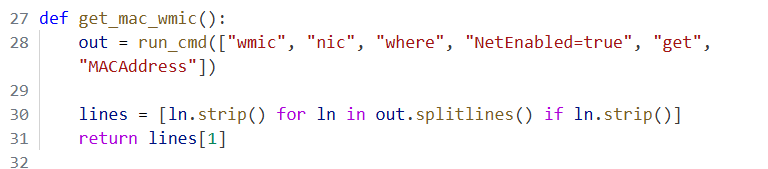


Создание программы:

Импортирование необходимых библиотек для взаимодействия с ОС Windows и создание переменной, которая хранит название файла с шифром:



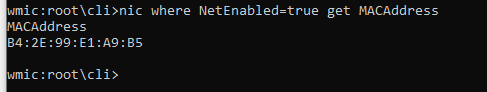
Далее была разработана функция, которая вычисляет MAC адрес моего ПК:



Записываем в переменную out вызов командной строки, которая должна вывести мак активного сетевого адаптера:



Проверка в cmd и пример того, что будет в переменной out:

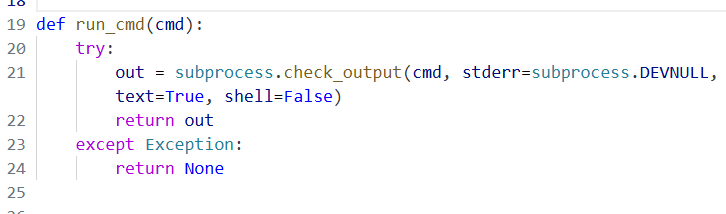


Затем в переменную lines заношу значение вызова, то что будет в lines:

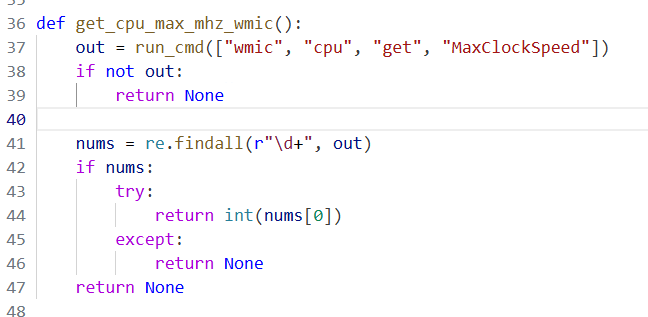


И функция возвращает 1 элемент массива, именно B4:2E:99:E1:A9:B5.

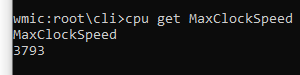
Также для работы с командной строкой использовалась функция, которая возвращала ответ cmd:



Далее проводим такую же процедуру для поиска частоты процессора:



Out в консоли выглядит так:



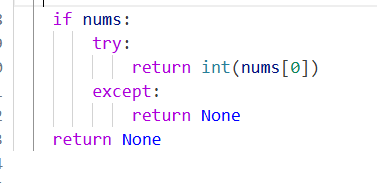
Далее ищем в строке цифру



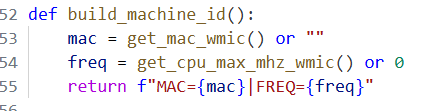
Re.findall вернет массив с 1 элементом, как раз частота:



Возвращаем первый элемент массива, если он не пуст:



После того, как я создал функции для получения MAC адреса и частоты CPU, я сформировал функцию, которая формирует некий id моего пк, как раз MAC+CPU speed:



Вызов вернет строчку:

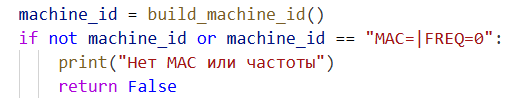


Теперь самое сложное, создание лицензии:

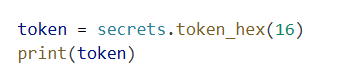
Функция создания:



Сначала вызывается ранее созданная функция, которая возвращает строчку с параметрами системы. Идет проверка, есть ли MAC и Частота:



Затем создается токен, для безовасности:



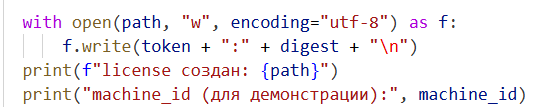
Например: 

Затем идет кеширование и токен сливается с параметрами MAC и частоты:



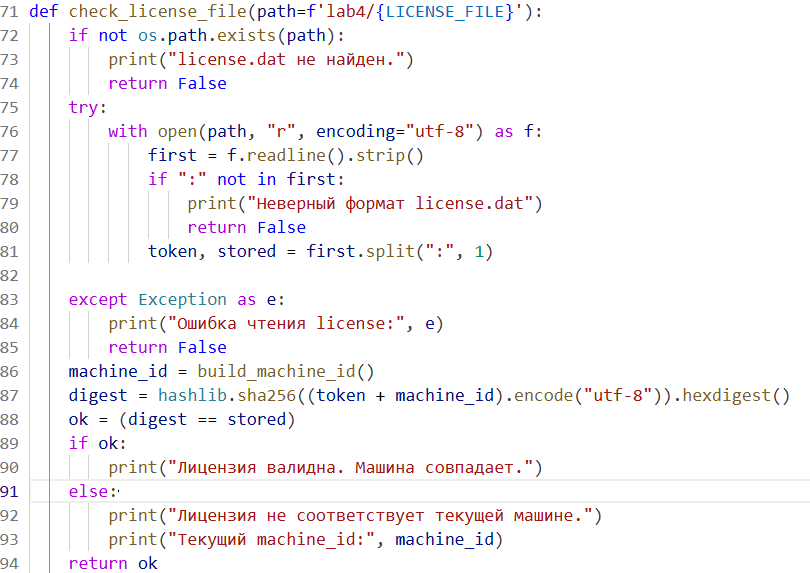
Получаем такой хеш: 

Который сохраняем в директорию проекта и выводим информацию:



Файл лицензии готов.

Далее была создана функция для проверки лицензии, одна декодирует данные из файла и сравнивает с текущими:



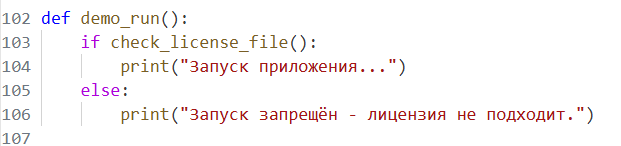
Я запустил функцию создания лицензии, затем функцию проверки:



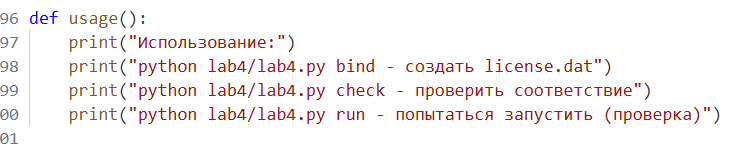
Затем для проверки я специально испортил данные в файле лицензии (изменил токен):



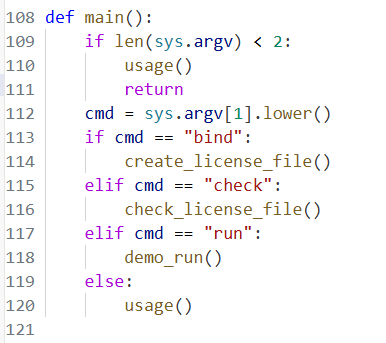
Далее я создал функцию, которая имитирует запуск приложения:



Осталось сделать пользовательскую логику, я создал функцию, которая выводит правила управления программой:



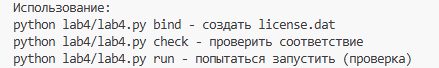
И функция main, которая обрабатывает команды пользователя:



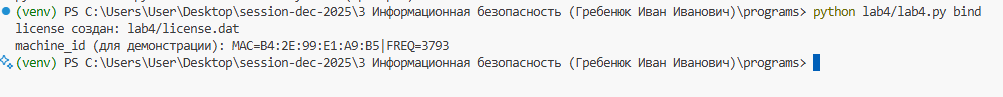
1. python binding\_windows.py bind ввод данной команды вызовет функцию для создания лицензии в директории;
2. python binding\_windows.py check ввод данной команды вызовет функцию проверки лицензии;
3. python binding\_windows.py run данная функция имитирует запуск программы, если есть доступ на текущем ПК – будет вывод в консоль: «Запуск приложения...»

**Результат работы приложения**

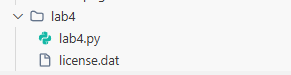
**Запуск без параметров, выводит правила:**

****

**Запуск python lab4/lab4.py bind:**



В директории появился файл лицензии:



В нем содержаться зашифрованные данные моей системы:

****

**Запуск python lab4/lab4.py check:**

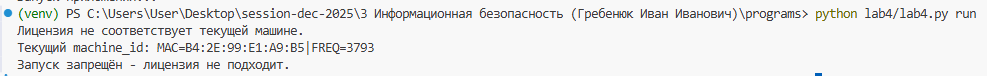
****

**Запуск python lab4/lab4.py run:**

**Приложение успешно запущено:**

****

**Для проверки меняю token в license.dat и повторяю команду python lab4/lab4.py run:**

****

**Выводы**

Выполнена лабораторная работа по реализации привязки программы к конкретному компьютеру с использованием MAC-адреса сетевой карты и максимальной тактовой частоты процессора.

Разработан алгоритм формирования идентификатора ПК, его защита с помощью token и SHA-256 и хранение в файле лицензии.

Реализованы режимы создания лицензии, проверки соответствия и демонстрационного запуска.

Показано, почему хранение хеша с токеном безопаснее, чем хранение явных параметров, и почему комбинация характеристик надёжнее одной.