**1.0 MAC адрес.**

МАС адрес - уникальный физический идентификатор устройства.

Введя команду ifconfig - увидим информацию про сеть, в частности МАС адрес, айпи и прочее.

Замена МАС адреса позволит:

1) Выдать своё устройство за чужое

2) Можно обойти фильтры, проверяющие МАС адреса

3) Анонимность в сети

**1.01 Смена МАС адреса в Linux:**

ifconfig

ifconfig eth0 down - необоходимо отключиться от сети

ifconfig eth0 hw ether 00:11:22:33:44:55 - меняем МАС. Опция ether указывает на МАС адрес

ifconfig eth0 up - подключаемся к сети с новым МАС адресом.

**1.1 Реализация на Python:**

subprocess.call - функция модуля subprocess, позволяющая выполнять команды

import subprocess # импортируем модуль subprocess

subprocess.call("ifconfig eth0 down", shell=True) # функция call исполняет команду, отключаемся от сети

subprocess.call("ifconfig eth0 hw ether 00:11:22:33:44:55", shell=True) # меням МАС на указанный, ether - указатель на МАС

subprocess.call("ifconfig eth0 up", shell=True) # подключаемся к сети

**1.2 Переменные для сетей и МАС адресов:**

import subprocess # импортируем модуль subprocess

interface = "eth0" - задаём переменной interface - eth0

new\_mac = "00:11:66:55:44:33" - задаём переменной new\_mac МАС адрес

print("Меням МАС адрес для " + interface + " на " + new\_mac)

subprocess.call("ifconfig " + interface + " down", shell=True) # функция call исполняет команду, отключаемся от сети

subprocess.call("ifconfig " + interface + " hw ether " + new\_mac, shell=True) # меням МАС на указанный, ether - указатель на МАС

subprocess.call("ifconfig " + interface + " up", shell=True) # подключаемся к сети

**1.3 Пользовательский ввод названия интерфейса и МАС адреса:**

import subprocess # импортируем модуль subprocess

interface = input("Введите имя интерфейса: ") - пользователь введёт имя в переменную

new\_mac = input("Укажите новый МАС: ") - пользователь введёт МАС в переменную

print("Меням МАС адрес для " + interface + " на " + new\_mac)

subprocess.call("ifconfig " + interface + " down", shell=True)

subprocess.call("ifconfig " + interface + " hw ether " + new\_mac, shell=True)

subprocess.call("ifconfig " + interface + " up", shell=True)

**1.3.5 Обрабатываем ввод пользователя:**

subprocess.call(["список"]) - данный метод позволит сделать некую защиту ввода, поскольку в переменную мы могли загнать любую линусовскую команду, которая была бы исполнена. Такой метод позволит интерпритировать переменную исключительно как объект, который не нужно исполнять, будь он назван командой, поскольку он является элементом списка.

subprocess.call(["ifconfig", interface, "down"])

subprocess.call(["ifconfig", interface, "hw", "ether", new\_mac])

subprocess.call(["ifconfig", interface, "up"])

**1.4 Парсер аргументов для программы:**

import subprocess # импортируем модуль subprocess

import optparse # импортируем модуль optparse

parser = optparse.OptionParser()

# OptionParser - класс модуля optparse. создаём экземпляр этого класса, объект - parser.

# parser - дочерний элемент, умеющий всё, что умеет его родитель - класс OptionPaser

# Создали объект, добавили опции, вызвали метод parse\_args, который возращает аргументы

# и значения. python MACchange1.py --interface etho0 --mac 00:00:00:11:11:11

parser.add\_option("-i", "--interface", dest="interface", help="Интерфейс для изменения МАС адреса")

parser.add\_option("-m", "--mac", dest="new\_mac", help="Новый МАС адрес")

(options, arguments) = parser.parse\_args()

# метод парс\_аргс возвращает параметры в переменные options, arguments. В аrguments будут храниться аргументы

# --interface, --mac. В переменной options будут храниться eth0, 00:00:00:11:11:11.

# Чтобы получить доступ к значению интерфейса, которое ввел пользователь, необходимо options.interface

# доступ к значению МАС, необходимо options.new\_mac

interface = options.interface

new\_mac = options.new\_mac

print("Меням МАС адрес для " + interface + " на " + new\_mac)

subprocess.call(["ifconfig", interface, "down"])

subprocess.call(["ifconfig", interface, "hw", "ether", new\_mac])

subprocess.call(["ifconfig", interface, "up"])

**1.5 Рефакторинг. Функции.**

import subprocess # импортируем модуль subprocess

import optparse # импортируем модуль optparse

def change\_mac(interface, new\_mac):

print("Меням МАС адрес для " + interface + " на " + new\_mac)

subprocess.call(["ifconfig", interface, "down"])

subprocess.call(["ifconfig", interface, "hw", "ether", new\_mac])

subprocess.call(["ifconfig", interface, "up"])

# Создали функцию change\_mac с параметрами interface и new\_mac.

parser = optparse.OptionParser() # OptionParser - класс модуля optparser

parser.add\_option("-i", "--interface", dest="interface", help="Интерфейс для изменения МАС адреса")

parser.add\_option("-m", "--mac", dest="new\_mac", help="Новый МАС адрес")

(options, arguments) = parser.parse\_args()

change\_mac(options.interface, options.new\_mac)

# вызываем функцию change\_mac c параметрами (interface, new\_mac)

**1.5.5 Функции**

import subprocess # импортируем модуль subprocess

import optparse # импортируем модуль optparse

def get\_arguments(): # получить аргумент

parser = optparse.OptionParser() # OptionParser - класс модуля optparser

parser.add\_option("-i", "--interface", dest="interface", help="Интерфейс для изменения МАС адреса")

parser.add\_option("-m", "--mac", dest="new\_mac", help="Новый МАС адрес")

return parser.parse\_args()

def change\_mac(interface, new\_mac):

print("Меням МАС адрес для " + interface + " на " + new\_mac)

subprocess.call(["ifconfig", interface, "down"])

subprocess.call(["ifconfig", interface, "hw", "ether", new\_mac])

subprocess.call(["ifconfig", interface, "up"])

(options, arguments) = get\_arguments()

change\_mac(options.interface, options.new\_mac)

**1.6 Проверка ввода с условными операторами**

import subprocess # импортируем модуль subprocess

import optparse # импортируем модуль optparse

def get\_arguments(): # получить аргумент

parser = optparse.OptionParser() # OptionParser - класс модуля optparser

parser.add\_option("-i", "--interface", dest="interface", help="Интерфейс для изменения МАС адреса")

parser.add\_option("-m", "--mac", dest="new\_mac", help="Новый МАС адрес")

(options, arguments) = parser.parse\_args()

if not options.interface:

parser.error("Укажите интерфейс. Для получения информации используйте --help")

elif not options.new\_mac:

parser.error("Укажите МАС адрес. Для получения информации используйте --help")

return options

def change\_mac(interface, new\_mac):

print("Меням МАС адрес для " + interface + " на " + new\_mac)

subprocess.call(["ifconfig", interface, "down"])

subprocess.call(["ifconfig", interface, "hw", "ether", new\_mac])

subprocess.call(["ifconfig", interface, "up"])

options = get\_arguments()

change\_mac(options.interface, options.new\_mac)