**Project 01**

**MAC Changer Tool**

**CS4061**

**Ethical Hacking Concepts and Practices**

**Submitted by:** Momenah Saif

**Roll number:** 21i-1909

Table of Contents

[**Windows** 3](#_Toc160919700)

[ **Code Details** 3](#_Toc160919701)

[ **Libraries and Running of Code** 4](#_Toc160919702)

[ **Output** 5](#_Toc160919703)

[ **Wireshark** 8](#_Toc160919704)

[**Linux** 9](#_Toc160919705)

[ **Code Details** 10](#_Toc160919706)

[ **Libraries and Running of code** 10](#_Toc160919707)

[ **Output** 10](#_Toc160919708)

[ **Wireshark** 10](#_Toc160919709)

# **Windows**

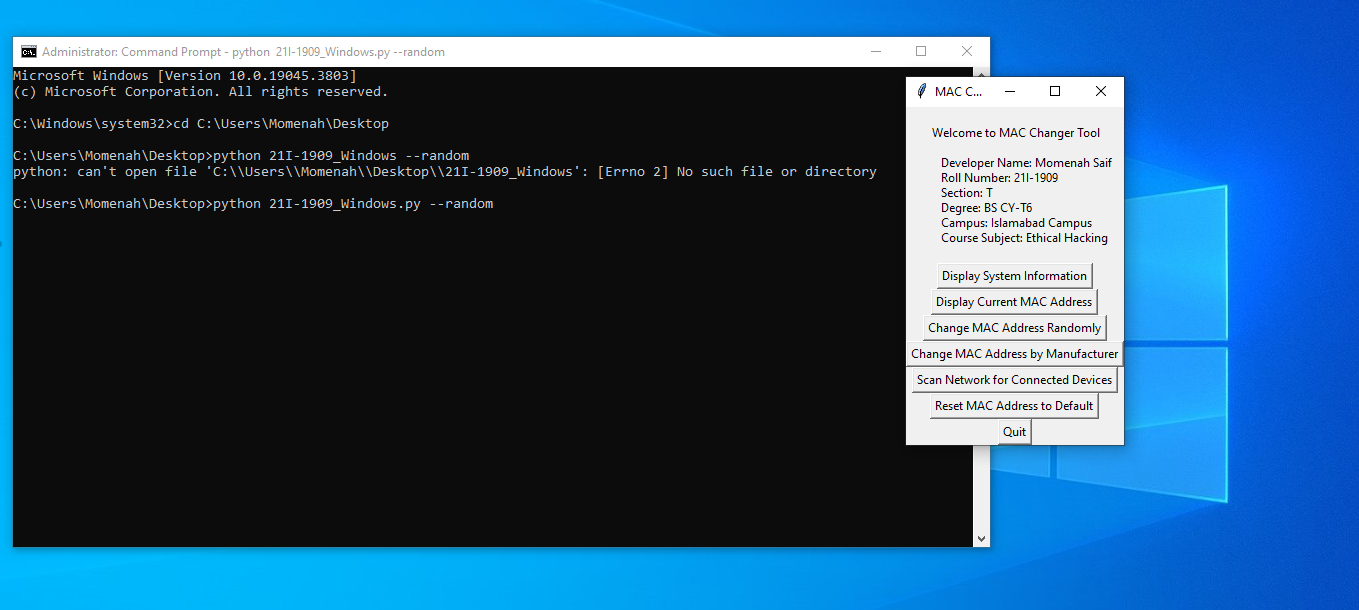
## **Code Details**

* + **Global constants:**
* network\_interface\_reg\_path: This defines the Windows registry path where network adapter configuration is stored.
* transport\_name\_regex: This is used to extract the adapter’s name from the output of the getmac command used.
* mac\_address\_regex: This expression is used to validate the format of a MAC address.
* get\_user\_adapter\_choice: This function retrieves a list of connected network adapters and their MAC addresses using getmac builtin function and prompts the user to choose one for further actions.
* **Global Functions:**
* change\_mac\_address: This function takes the adapter name and the new MAC address as input. It modifies the Windows registry to update the MAC address of future packets and returns the adapter index.
* get\_random\_mac\_address: This function generates a random hex number as MAC address following a specific format.
* disable\_adapter and enable\_adapter: These functions use built-in subprocess to execute system commands for disabling and enabling the network adapter respectively to ensure that the MAC has been changed. These functions restart the adapter after MAC change.
* **Class, constructor and its functions**
* This class defines the main application logic for the GUI.
* The constructor (\_\_init\_\_) creates the main window, sets its name, and adds various buttons and labels using tkinter widgets. Each button is linked to a corresponding function in the class for a specific action.
* Functions like display\_system\_info and display\_current\_mac update the label in the window with relevant information. display\_system\_info uses datetime to display date and time of system. display\_current\_mac calls get\_connected\_adapters\_mac\_address() and displays the mac
* Functions like change\_mac\_random and change\_mac\_by\_manufacturer handle changing the MAC address based on user selection. They involve getting user input, calling get\_random\_mac\_address or retrieving manufacturer MAC addresses from links, and using change\_mac\_address to update the registry.
* scan\_network\_devices attempts to use subprocess to execute the arp -a command and display the ARP cache (network devices) on the label.
* reset\_mac resets the MAC address to its default value by calling change\_mac\_address with an empty string.
* get\_manufacturers function seems intended to scrape MAC addresses from website based on manufacturer name.
* fetch\_mac\_table(self, url) function downloads the content of the provided URL using the requests library.It returns the downloaded text if successful (status code 200), otherwise returns None.
* parse\_mac\_table(self, text) function initializes an empty dictionary mac\_table.Splits the text by lines and iterates through them. For each line with at least two parts (separated by tabs), adds an entry to the dictionary: key is the first part, value is the second part.
* search\_mac\_by\_manufacturer(self, mac\_table, manufacturer) function initializes found\_mac to None.Iterates through key-value pairs in mac\_table.Checks if the manufacturer name (converted to lowercase) is present (lowercase) in the value .If found, sets found\_mac to the key and breaks the loop.It returns the found\_mac address or None if not found.
* complete\_mac(mac) function pads the provided MAC address string with colons and additional hexadecimal characters to reach a total length of 12.

## **Libraries and Running of Code**

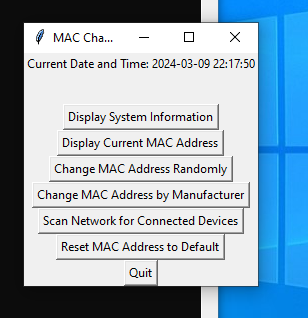
* tkinter (as tk): This is the core library for creating graphical user interfaces in Python. It provides widgets like buttons, labels, and text boxes for making better interface for the tool.
* tkinter. simpledialog: This submodule of tkinter offers functions for creating simple dialog boxes like popups for user input.It is used to enter the name of manufacture.
* tkinter. messagebox: Another submodule of tkinter used for displaying message boxes with icons and buttons for informing the user. Used to tell output of manufacturer function.
* subprocess: This library allows you to execute system commands from your Python code and capture their output.
* random: This library provides functions for generating random numbers, which is used here to create random MAC addresses.
* re: This library provides regular expression functionality for searching and manipulating text based on patterns.
* requests: This library helps make HTTP requests to websites and retrieve their content.
* string: This module provides various string manipulation functions.
* datetime: This module helps with working with dates and times.
* bs4:This library is used for parsing HTML content, which is used here to scrape data from websites to populate the manufacturer MAC address list.
* **Running**

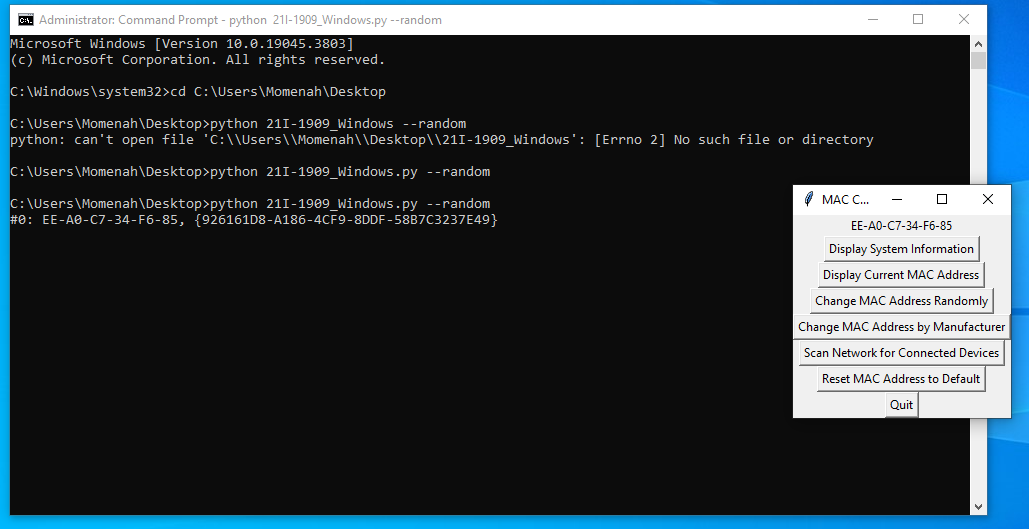
Opened the cmd in admin privileges and then went to the directory of .py file and wrote command.

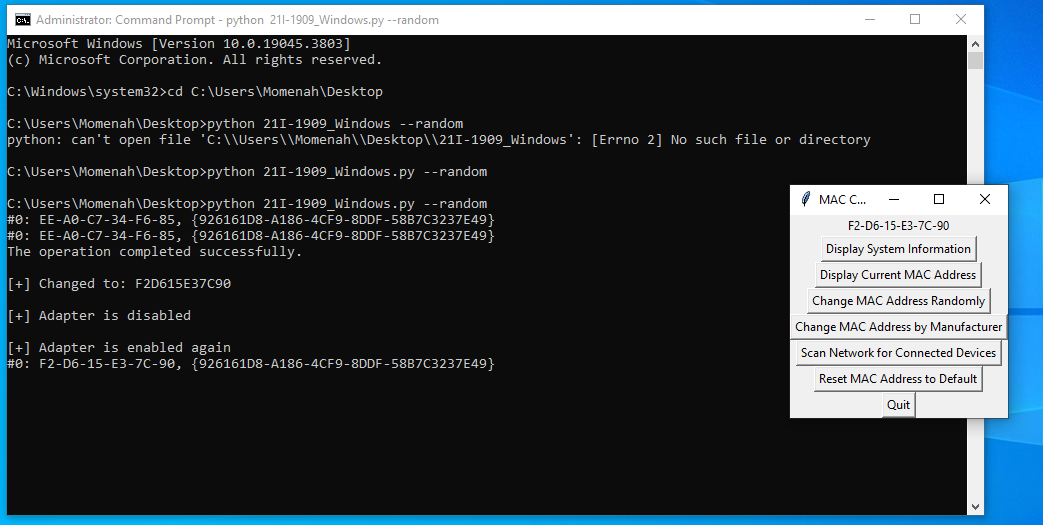


## **Output**

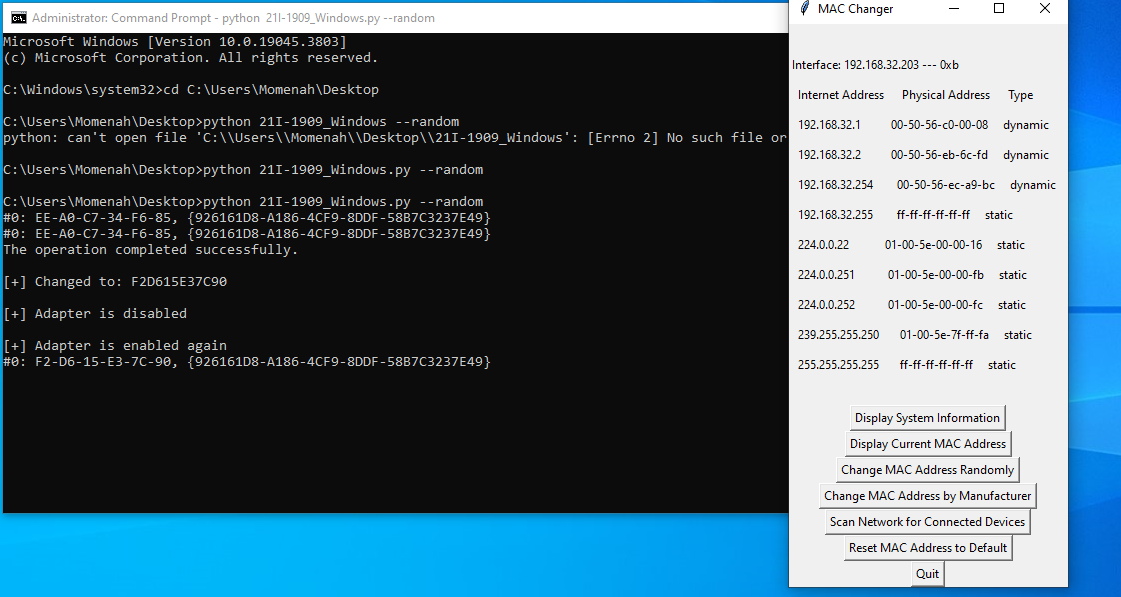
* **System Information**



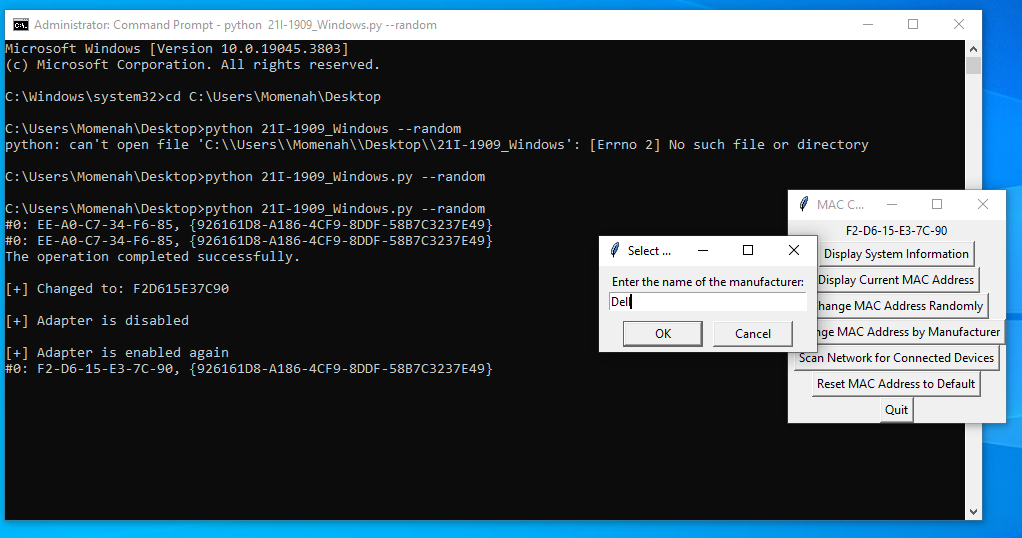
* Current MAC
* Change MAC Randomly



* Scan Network

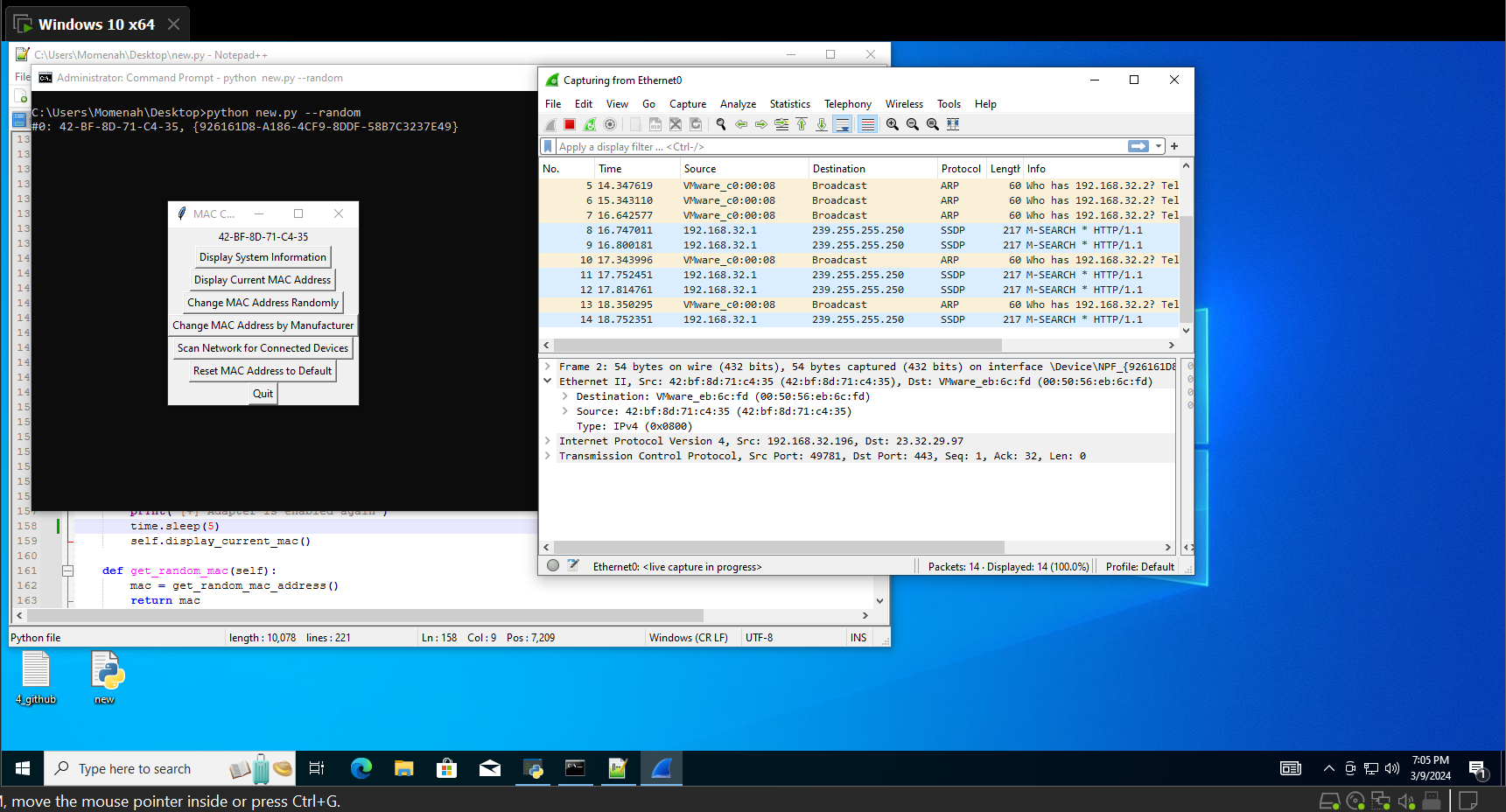


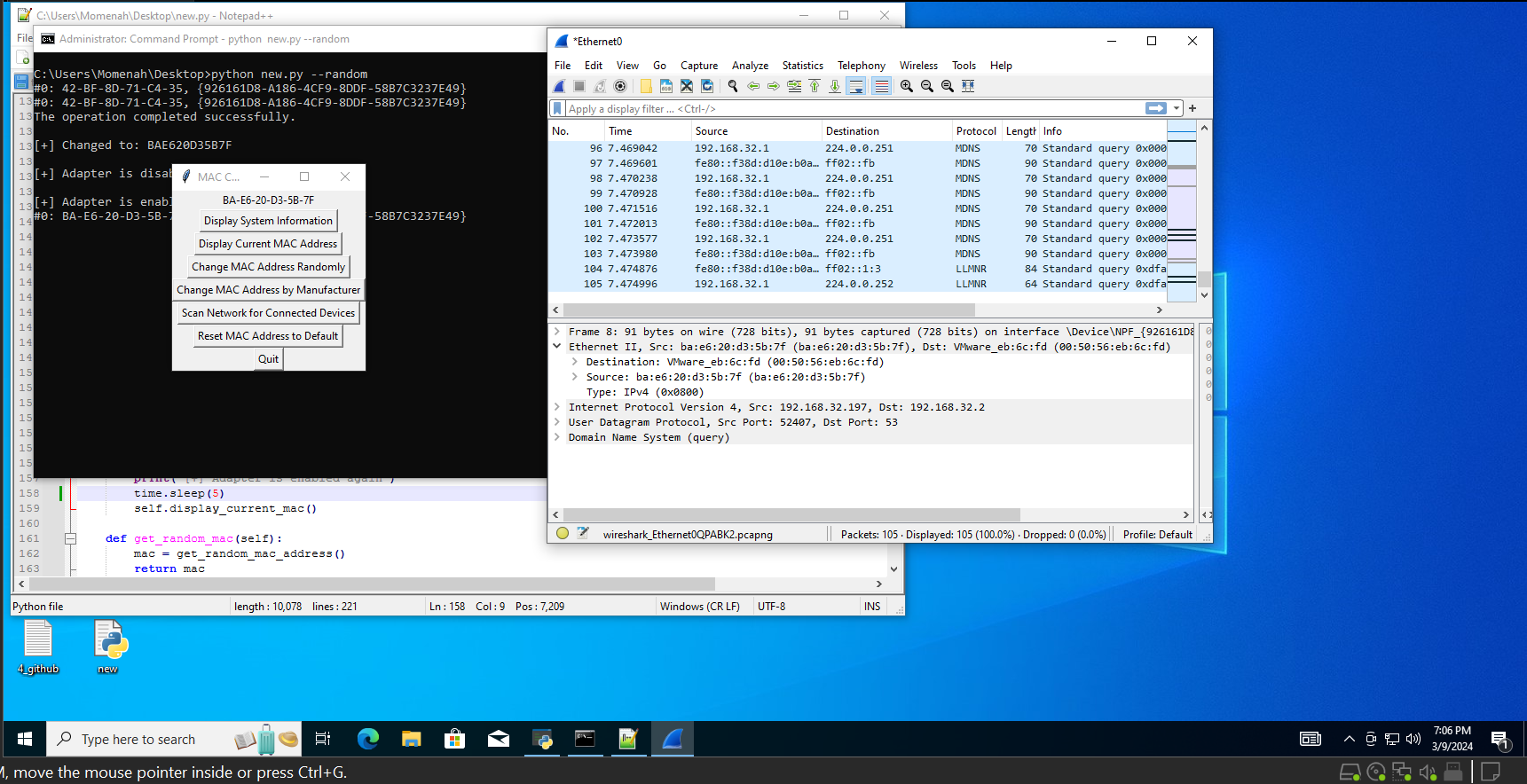
* Change MAC by Manufacturer



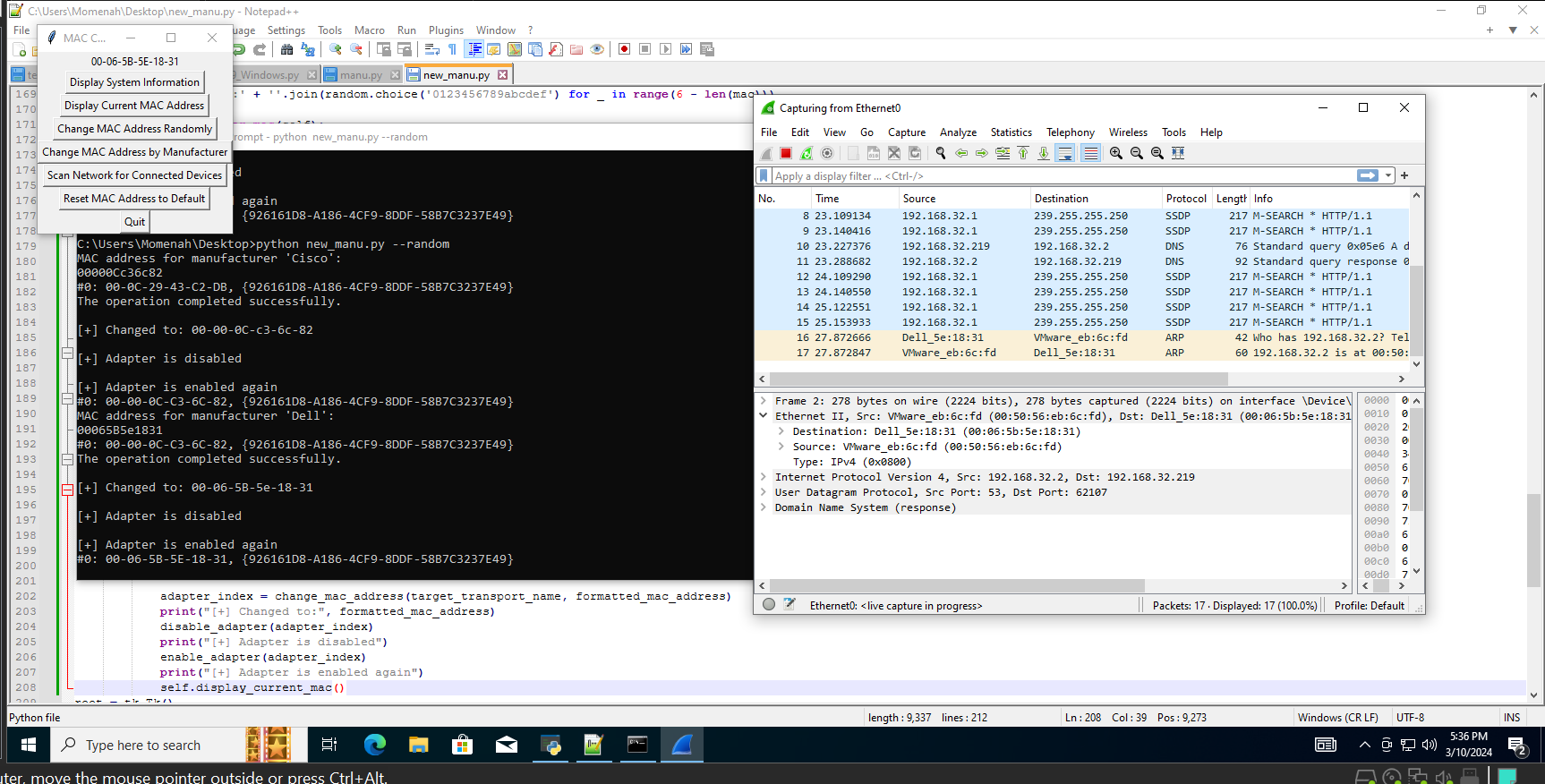
## **Wireshark**

* Random Change



****

* Change by name of Manufacturer



# **Linux**

## **Code Details**

* **Class**

This class defines the main functionalities of the application.It initializes the Tkinter root window and sets the title as "MAC Changer".It creates various buttons with labels for the functionalities:

1. Display System Information
2. Display Current MAC Address
3. Change MAC Address Randomly
4. Change MAC Address by Manufacturer
5. Scan Network for Connected Devices
6. Reset MAC Address to Default
7. Quit

Each button click triggers a corresponding function defined within the class.

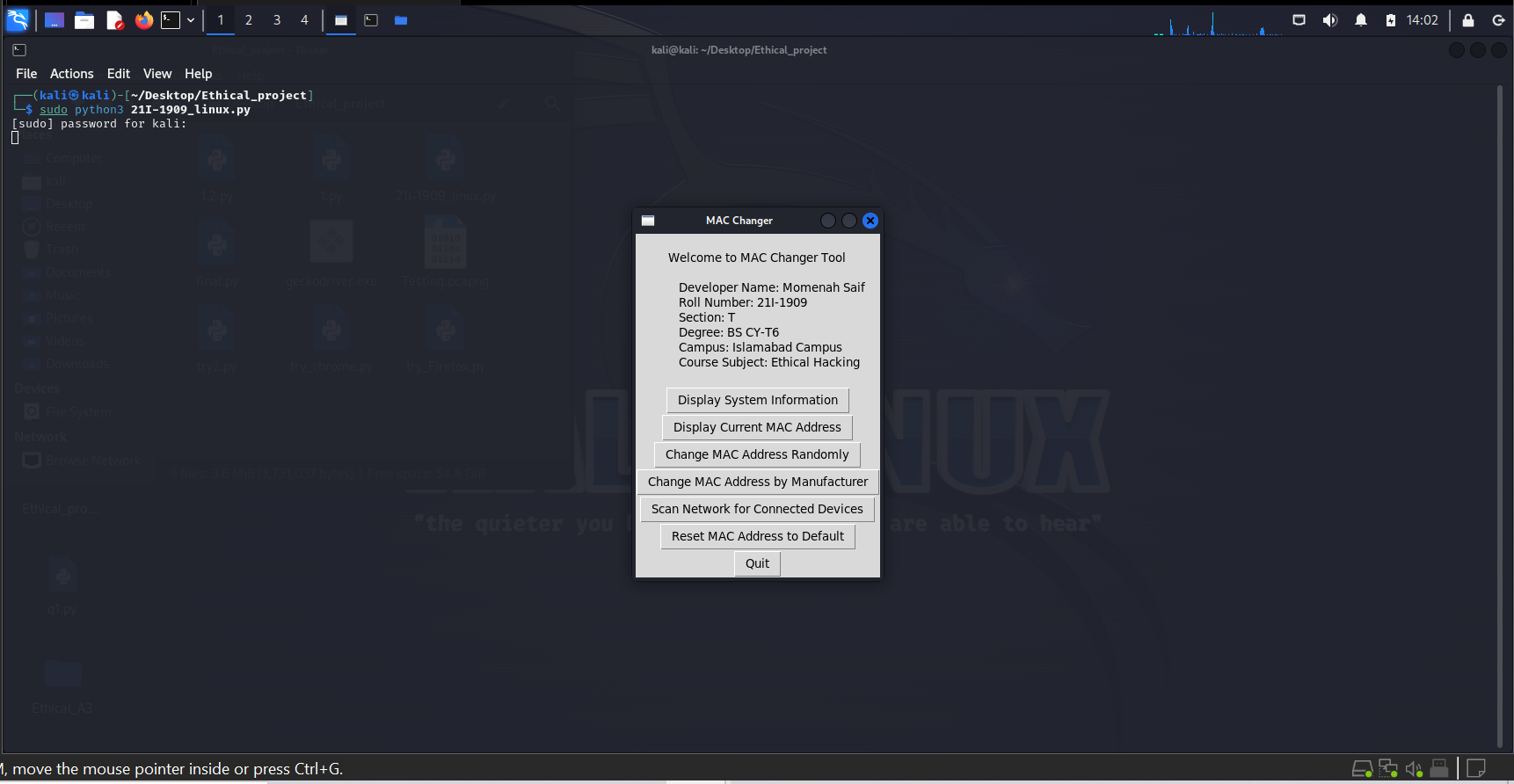
* **Functions**
  + display\_system\_info: This function uses subprocess to execute the date command and displays the output on the label.
* display\_current\_mac: This function uses netifaces to get a list of network interfaces and iterates through them. It retrieves the MAC address for each valid interface. The retrieved information is then displayed on the label.
* change\_mac\_random: This function iterates through network interfaces, It uses subprocess to bring the interface down, generate a random MAC address using random, change the MAC address using macchanger , and finally bring the interface back up.
* change\_mac\_by\_manufacturer: This function first retrieves a dictionary of manufacturer MAC address prefixes using the get\_manufacturers function. It then prompts the user for a manufacturer name using simpledialog. If the manufacturer is found in the retrieved data, it uses subprocess to bring the interface down, set the MAC address based on the manufacturer prefix, and bring the interface back up.
  + scan\_network\_devices: This function uses subprocess to execute the arp -a command and displays the output on the label. This command shows the MAC addresses of devices connected to the network.
  + reset\_mac: This function iterates through network interfaces. It attempts to reset the MAC address to its default using macchanger and subprocess. Any errors encountered are displayed for the specific interface.
  + get\_random\_mac: This function generates a random MAC address by creating a list of 12 random hexadecimal digits and joining them with colons.
  + get\_manufacturers: This function is responsible for collecting manufacturer MAC address prefixes. It retrieves data from multiple websites containing such information using requests and parses the HTML content using BeautifulSoup. The extracted manufacturer and MAC address prefix pairs are stored in a dictionary and returned.
* fetch\_mac\_table(self, url) :Downloads the content of the provided URL using the requests library.It returns the downloaded text if successful (status code 200), otherwise returns None.
* parse\_mac\_table(self, text): Initializes an empty dictionary mac\_table.Splits the text by lines and iterates through them. For each line with at least two parts (separated by tabs), adds an entry to the dictionary: key is the first part, value is the second part.
* search\_mac\_by\_manufacturer(self, mac\_table, manufacturer) : Initializes found\_mac to None.Iterates through key-value pairs in mac\_table.Checks if the manufacturer name (converted to lowercase) is present (lowercase) in the value .If found, sets found\_mac to the key and breaks the loop.It returns the found\_mac address or None if not found.
* complete\_mac(mac) :Pads the provided MAC address string with colons and additional hexadecimal characters to reach a total length of 12.

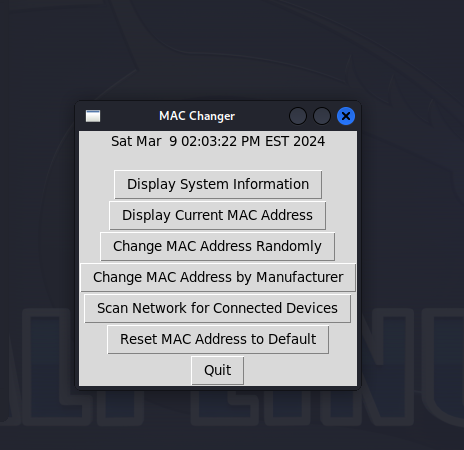
## **Libraries and Running of code**

* + - tkinter: This is the core library used for creating the GUI elements like buttons and labels.
    - simpledialog (from tkinter): This provides functions for displaying simple dialog boxes like for user input.
    - messagebox (from tkinter): This is used for displaying message boxes for notifications or errors.
    - subprocess: This library allows the program to execute shell commands on the operating system. This is necessary for functionalities like changing the MAC address as it requires administrative privileges.
    - netifaces: This library helps interact with network interfaces on the system. It's used to retrieve and manipulate network interface information like the MAC address.
    - random: This library provides functions to generate random numbers. It's used in this case to generate random MAC addresses.
    - requests: This library enables making HTTP requests to websites. It's used to scrape data from websites containing manufacturer MAC address prefixes.
    - BeautifulSoup: This library helps parse HTML content retrieved from websites. It's used to extract manufacturer and MAC address information from the scraped websites.
* **Running**

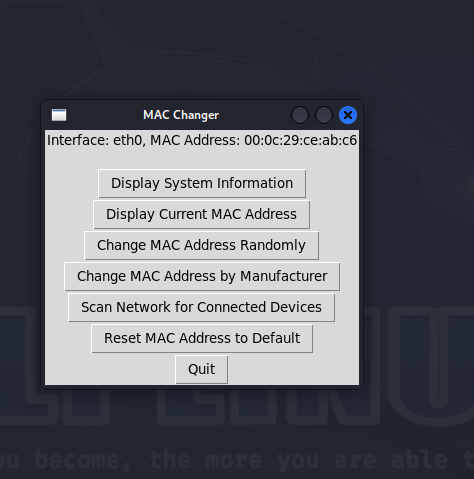
Opened terminal where the file was present and ran sudo python3 21I-1909\_linux.py

## **Output**

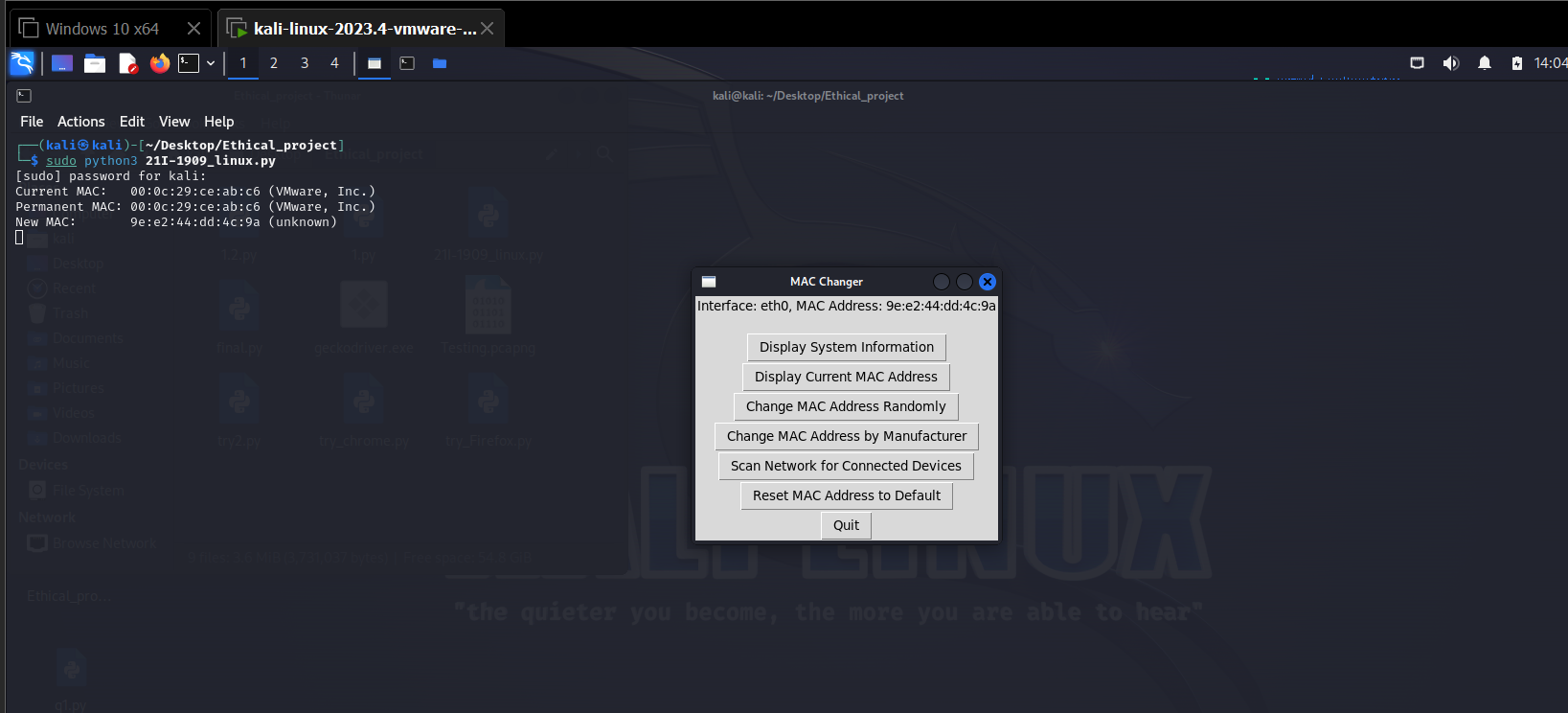
* + Developer info
  + System Info



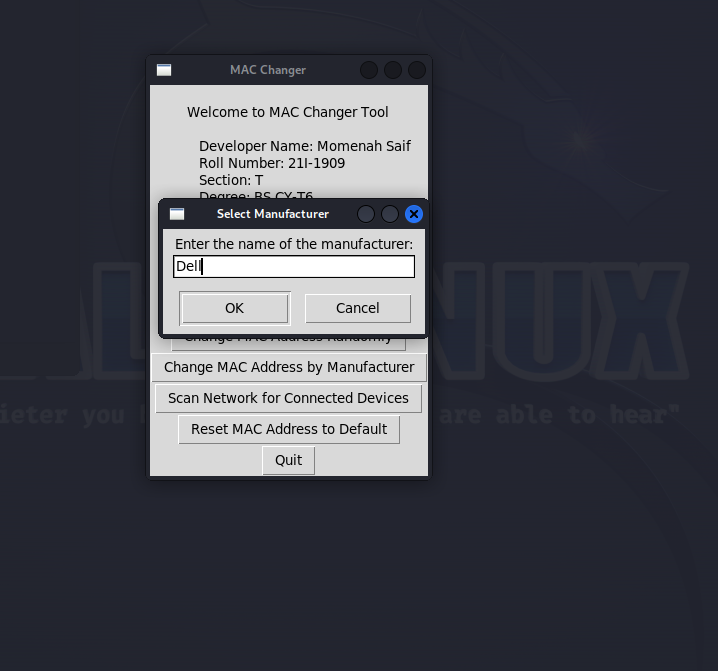
* Current MAC



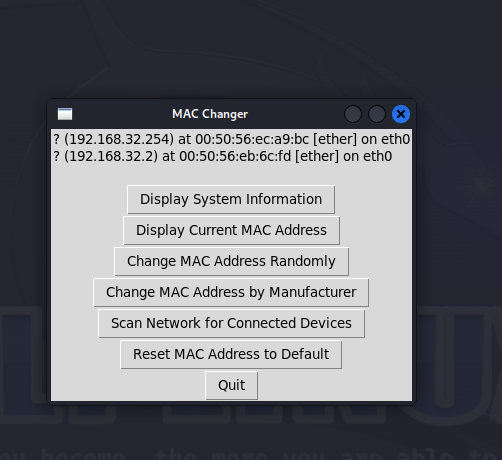
* Change MAC Randomly



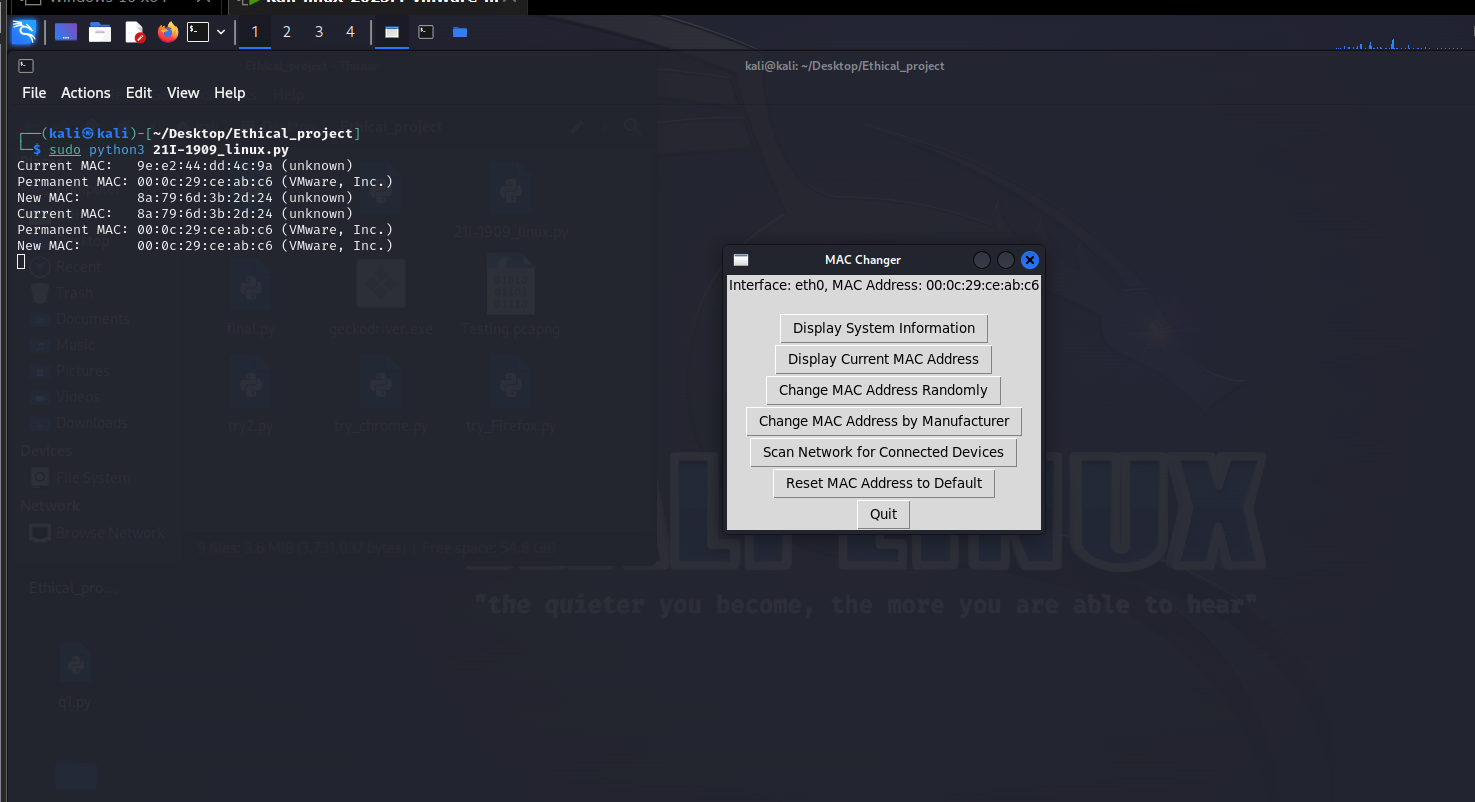
* Change MAC by Manufacturer



* Scan Network

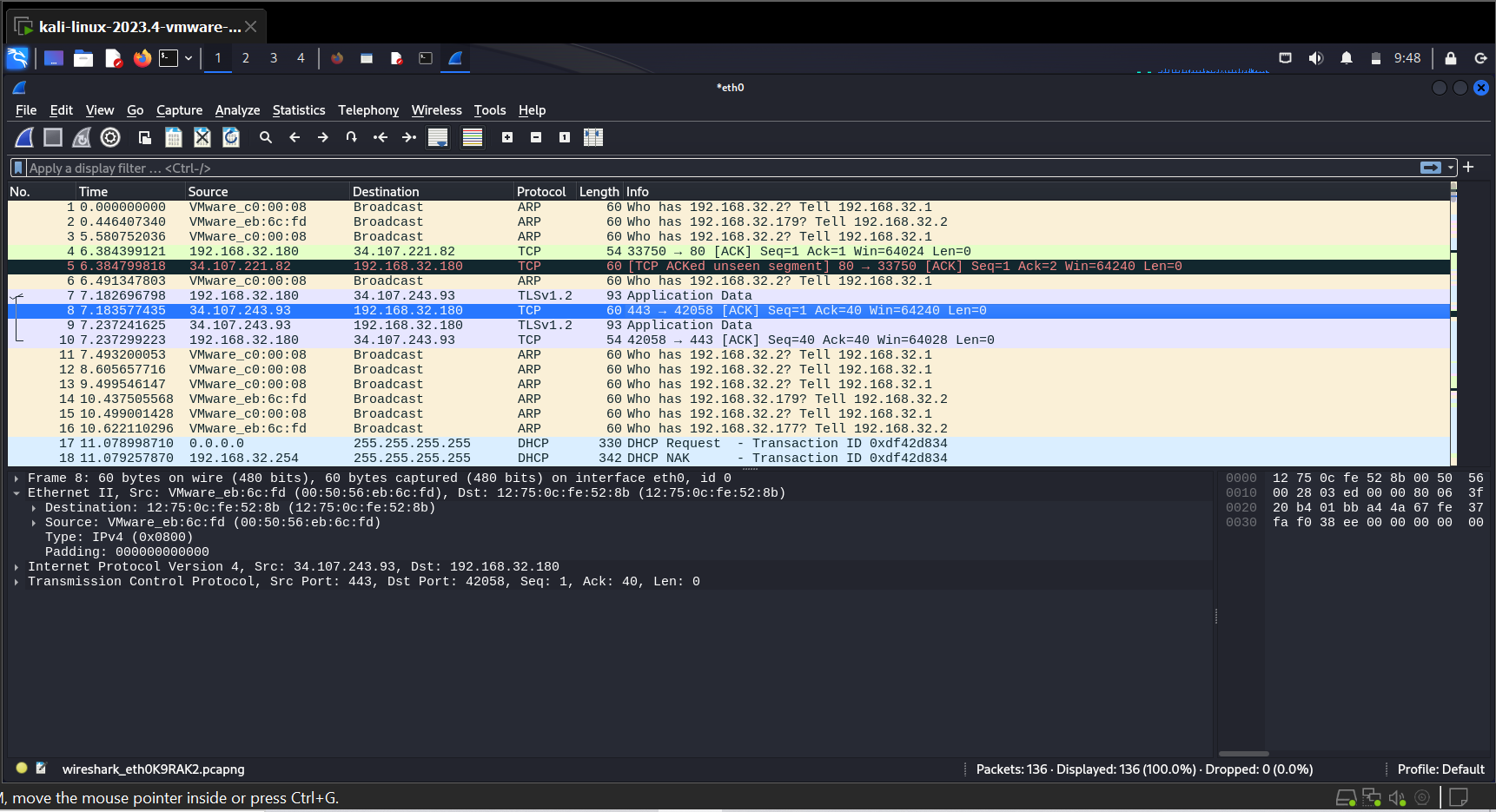


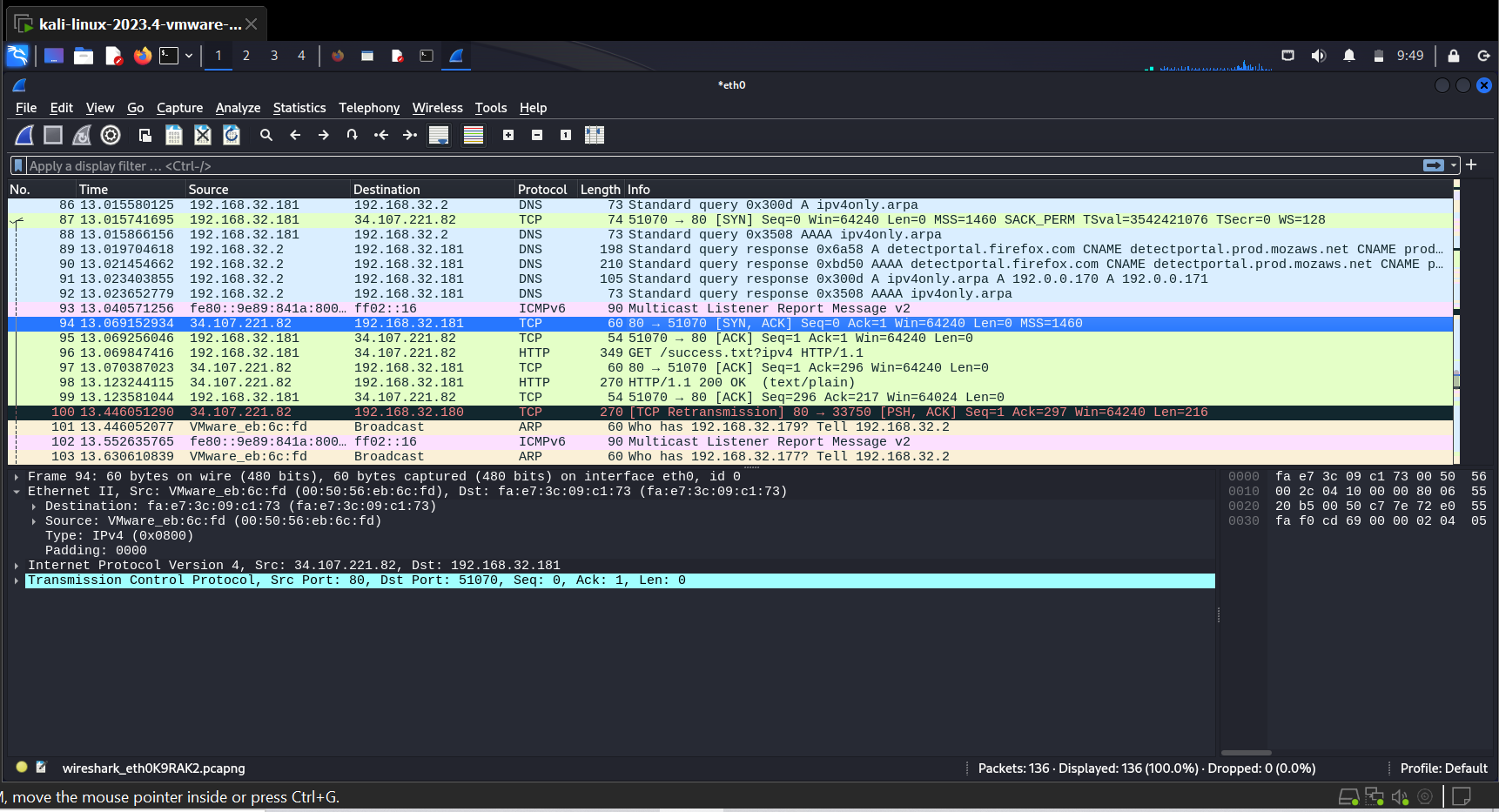
* Reset MAC



## **Wireshark**

* Change Randomly





* Change by Manufacturer

