DEF GET\_MAC():

1. we declare function **get\_mac(ip)** which requires a parameter **ip**.
2. we then broadcast the destination MAC address to ensure that all packets are transmitted to all nodes in a subnet
3. we combine the two packets to form a fully ARP request broadcast packet that will be sent to all the nodes in the subnet.
4. In the **srp()** function, we set the packet we created as argument,
5. timeout argument is set to 3 second. This means that the packet will wait 3 seconds for response before moving on to the next node.
6. Next we set the verbose argument to false, to limit the information outputted by the function to only the most relevant.
7. The function **srp()** return two lists. The first list contains the sent and the answered packets. The second list contains the unanswered packets. Now we set index to zero, [0] after a call to the function to make sure only the first list is assigned to variable **answered\_list**.
8. we return from the first list and the second list (which is the answered packet), we return the **hwsrc** field which is the MAC address of the source packet.

Def Process():

1. uses **if** statement to check if our packet has layer scapy.ARP using the haslayer function, and also if the **op** field of layer scapy.ARP of our packet is equivalent to 2. This signifies a response packet.
2. assigns the variable **real\_mac**, with the value returned by the function **get\_mac()**and we passed the source ip (psrc) of our packet as an argument. This will get the actual MAC address of the machine sending the packet.
3. initializes variable response\_mac and assigns it with MAC address contained in the packet using the **hwsrc** field.
4. compares the real\_mac with the response \_mac and if they are not equal, that means the response\_mac might have been spoofed!
5. except **Index Error**: - **Index Error** might occur when our python script tries evaluating a packet, that is not destined to our machine. So we except this error, and we asked the program to carry on with the code execution and ignore that error!

**Try block of main()**

Here we import the sys and take the input from system using sys.argv[1] and assign it to iface

**In the except Inderxerror**

The conf.iface is used to get the iface value if the user doesn’t enter any interface value

**SNIFF FUNCTION**

1. defined a function called sniff
2. This function **sniff**will sniff the packets passing through our interface. The iface argument of function sniff specifies the interface to sniff on. The argument store is use to specify whether our computer should store the packets in memory or otherwise. In this case, we don’t want to store any packet, so we specify the value of store as **false**. Argument prn is use to specify a callback function. So function sniff will first run, sniff packets and call the callback function and passed the packets to it. Argument prn is assigned with function **process** which in this case is our callback function.