בתרגיל זה נרצה לזהות מצב שבו אנחנו תחת מתקפת Arp-spoofing ונרצה "להשתחרר" מאותה מתקפה.
השלב הראשון הוא קודם כל לזהות שמישהו תוקף אותנו ולגלות מי זה ומה כתובת הmac שלו ואחר מכן נוכל לפעול בדכים שנות על מנת לחסום אותו מלהמשיך לתקוף אותנו.

הארגומנט שנרצה לקבל הוא ה-Interface שאנחנו חושדים שנתקף
וכעת נשתמש בפונקציה מובנת של scapy הפונקציה sniff.

Sniff (store = false, prn=process, iface=iface)

Store יוגדר להיות F על מנת שיוכל להמשיך לרוץ כל הזמן. Prn זה תהליך שנרצה לבצע תוך כדי ההסנפה.

כעת נממש את process

```
def process(packet):
# if the packet is an ARP packet
if packet.haslayer(ARP):
    # if it is an ARP response (ARP reply)
    if packet[ARP].op == 2:
        try:
            # get the real MAC address of the sender
            real_mac = get_mac(packet[ARP].psrc)
            # get the MAC address from the packet sent to us
            response_mac = packet[ARP].hwsrc
            # if they're different, definitely there is an attack
            if real_mac != response_mac:
                print(f"[!] You are under attack, REAL-MAC: {real_mac.upper()}, FAKE-MAC: {response_mac.upper()}")
        except IndexError:
            # unable to find the real mac
            # may be a fake IP or firewall is blocking packets
            pass
```

ע"י הסנפה של פקטות אנחנו ננסה לזהות האם יש הבדל כלשהו בין הכתובות כאשר מתקבלות פקטות
נרצה לבדוק מה כתובת הmac האמיתית של השולח ע"י בקשת ARP, לשם כך נעזר בפונקציה שכתבנו בתרגיל הקודם לצורך זיהוי ה-MAC

```
def get_mac(ip):
    """
    Returns the MAC address of `ip`, if it is unable to find it
    for some reason, throws `IndexError`
    """
    p = Ether(dst='ff:ff:ff:ff:ff:ff')/ARP(pdst=ip)
    result = srp(p, timeout=3, verbose=False)[0]

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

כעת שאנחנו יודעים שמתקיפים אותנו ננסה להגן על עצמנו מפני המתקפה, נוכל לעשות זאת בכמה דרכים אחת האופציות שעלו הייתה לתקוף בחזרה אך פתרון אחר שעלה הוא פשוט לכבות ולהדליק את הפורט, ואז כל חיבור קיים עם התוקף ינותק ולאחר מכן לחסום את התוקף מליצור חיבורים נוספים בעתיד

```
def defeance(ip_a):

    # Restart the network interface with the specified name
    os.system('ip link set eth0 down')
    os.system('ip link set eth0 up')
    # Create a new socket
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Bind the socket to the IP address of the computer you want to block
    sock.bind(ip_a)

    # Listen for incoming connections on the socket
    sock.listen()

    # Accept any incoming connections and immediately close them
    while True:
        conn, addr = sock.accept()
        conn.close()
```