1-MAC-Spoofing

Name	Email
Omar Ashraf ElSayed Younes Abulila	omarabolilaofficial@gmail.com

Task Solution Source Code
https://github.com/omarafal/we-innovate-tasks/tree/main/1-MAC-Spoofer

Task

Create a script that changes the MAC address every three minutes and then resets it upon exit.

Task Solution Overview

NOTE: The script is assumed to run inside a Linux machine. If run on any other machine, it will always fail.

A typical MAC address consists of 6 bytes and (on Linux) is stored at `/sys/class/net/[INTERFACE]/address`.

The script takes in one argument; the targeted interface.

Setting the first byte to `0x02` indicates that this MAC address is **locally administered** and we can use it to spoof. The remaining 5 bytes are randomly chosen numbers between `0x00` and `0xFF`.

Before setting the new random address, we have to first safely store away our original MAC address so as not to lose it and then restore it once the user is done using the script.

Task Solution Explanation

```python
from random import randint
import time
import sys
import os

def rand_mac():
    final_addr = "02"

    for _ in range(0, 5):
        final_addr += f":{randint(0, 255):02X}"

    return final_addr
```

The first four lines import the necessary libraries to generate random numbers, sleep for a specified time, grab command line arguments and to finally execute commands.

The function `rand_mac` generates the 5 random bytes and appends them to the fixated `0x02` byte and then returns the new address.

Now the `main` function:

```python
def main(intf):
    try:
        with open(f"/sys/class/net/{intf}/address") as file:
            org = file.read().strip()

        # store in temp file just in case
        with open("./spoof.tmp", "w") as file:
            file.write(org)
    except:
        print("This is not a valid interface.\n")
        sys.exit(1)
```

This first part opens the file where the MAC address is stored and grabs the original one. It then stores it in a temporary file just in case something went wrong like the terminal suddenly closing. That way if anything happens, the user still has a copy of the original MAC address.

This temporary file gets deleted after the script is done executing successfully.

```
try:
    while True:
        new_add = rand_mac()

        os.system(f"sudo ip link set dev {intf} down")
        os.system(f"sudo ip link set dev {intf} address {new_add}")
        os.system(f"sudo ip link set dev {intf} up")

        print("\nMAC address changed.\n")
        print(f"Running: ip link show {intf}")
        os.system(f"ip link show {intf}")
        time.sleep(60*3)
```

This code inside the `main` function is responsible for setting the new MAC address that is returned from the `rand_mac` function then printing out the information regarding the provided interface to prove that it changed.
Finally, it sleeps for 3 minutes before re-doing all of this again; setting a new random MAC address.

```
except KeyboardInterrupt:
    print("\nCleaning up before exiting.\nResetting MAC address.")

    try:
        os.system(f"sudo ip link set dev {intf} down")
        os.system(f"sudo ip link set dev {intf} address {org}")
        os.system(f"sudo ip link set dev {intf} up")
    except:
        print("Something went wrong.\n")
        sys.exit(1)
    # to protect the temp file just in case
    if os.path.exists("./spoof.tmp"):
        os.remove("./spoof.tmp")
```

This final major piece of code is responsible for cleaning up when the user hits `Ctrl+C` to stop the script.
The "cleanup" consists of resetting the MAC address to its original one as well as deleting the temporary file that contained it since it won't be necessary to keep if the original address was set successfully.

```
if __name__ == "__main__":
    try:
        intf = sys.argv[1] # grab interface
    except IndexError:
        print("Usage: python macspoof.py [INTERFACE]\n")
        sys.exit(1)

    main(intf)
```

At last, the user receives a message if no arguments were passed to the script at all to show the user how to correctly use the script.