IP and MAC Spoofing with Tor Integration

Imteyaz Ali

Tata Strive

Course: Cybersecurity Analyst

Instructor: Kaustubhamani Gothivareker

Submission Date: 5th February 2025

Abstract

This project demonstrates the process of IP and MAC address spoofing on a Kali Linux machine using macchanger and ifconfig utilities, alongside integration with the Tor network to enhance privacy and security. The goal was to develop a script that generates 1000 random IP and MAC addresses, changing them every 10 seconds. The use of Tor ensures anonymous communication, making it ideal for enhancing online privacy. The script was tested successfully, showing random IP and MAC addresses with Tor-enabled browsing. This report outlines the tools, methodology, results, and advantages of this spoofing technique.

Contents

Chapter 1

Introduction

1.1 Background

1.1.1 IP and MAC Address Spoofing

IP Spoofing refers to the act of modifying the source IP address in a packet's header to make it appear as though the packet is coming from a different source. This can be used for various purposes, including bypassing network filters, masking the identity, and performing denial of service attacks.

MAC Spoofing refers to the modification of the hardware address (MAC address) of a network interface card (NIC) in a computer. This can help bypass network restrictions, prevent tracking, and provide anonymity.

1.1.2 Tor Network

Tor (The Onion Router) is a privacy-focused, open-source network that helps users maintain anonymity by routing internet traffic through a series of volunteer-operated servers. This approach encrypts the user's data multiple times, making it difficult to track or trace the traffic back to its origin. Using Tor improves privacy by hiding the real IP address and enabling access to restricted content.

1.2 Objective of the Project

The main objectives of this project are:

- To demonstrate IP and MAC address spoofing using Kali Linux tools
- To create a Python script automating address changes every 10 seconds
- To integrate Tor network for anonymous communication
- To verify functionality and anonymity through testing

Chapter 2

Tools and Technologies Used

Kali Linux Penetration testing distribution with security tools

macchanger Linux utility for MAC address modification

Tor Onion routing network for anonymous communication

Python Scripting language for automation

curl Command-line tool for data transfer

Chapter 3

System Setup and Configuration

3.1 System Prerequisites

Install required packages:

```
sudo apt update
sudo apt upgrade
sudo apt install macchanger tor curl
```

3.2 Tor Configuration

Start Tor service and verify connection:

```
sudo systemctl start tor
curl --socks5 127.0.0.1:9050 http://icanhazip.com
```

Chapter 4

Methodology

4.1 IP and MAC Spoofing Script

The Python script performs these key functions:

- Generates random IP and MAC addresses
- Changes network configuration using system commands
- Implements 10-second interval changes
- Verifies Tor connectivity

4.2 Script Code

Listing 4.1: IP/MAC Spoofing Script

```python
import os
import random
import time
import subprocess

def generate_mac():
    return "00:11:22:%02x:%02x:%02x" % (
        random.randint(0, 255),
        random.randint(0, 255),
        random.randint(0, 255))

def generate_ip():
    return f"192.168.{random.randint(0,255)}.{random.randint(0,255)}"

def change_mac_ip():
    new_mac = generate_mac()
    new_ip = generate_ip()
    # MAC change commands
    os.system("sudo ifconfig eth0 down")
    os.system(f"sudo macchanger -m {new_mac} eth0")
    os.system("sudo ifconfig eth0 up")
    # IP change commands
    os.system(f"sudo ifconfig eth0 {new_ip} netmask 255.255.255.0")
    print(f"Changed to MAC: {new_mac}, IP: {new_ip}")

def test_tor():
    result = subprocess.run(
        ["curl", "--socks5", "127.0.0.1:9050", "http://icanhazip.com"],
        capture_output=True, text=True)
    print(f"Tor IP: {result.stdout.strip()}")

for _ in range(1000):
    change_mac_ip()
    test_tor()
    time.sleep(10)
```

Chapter 5

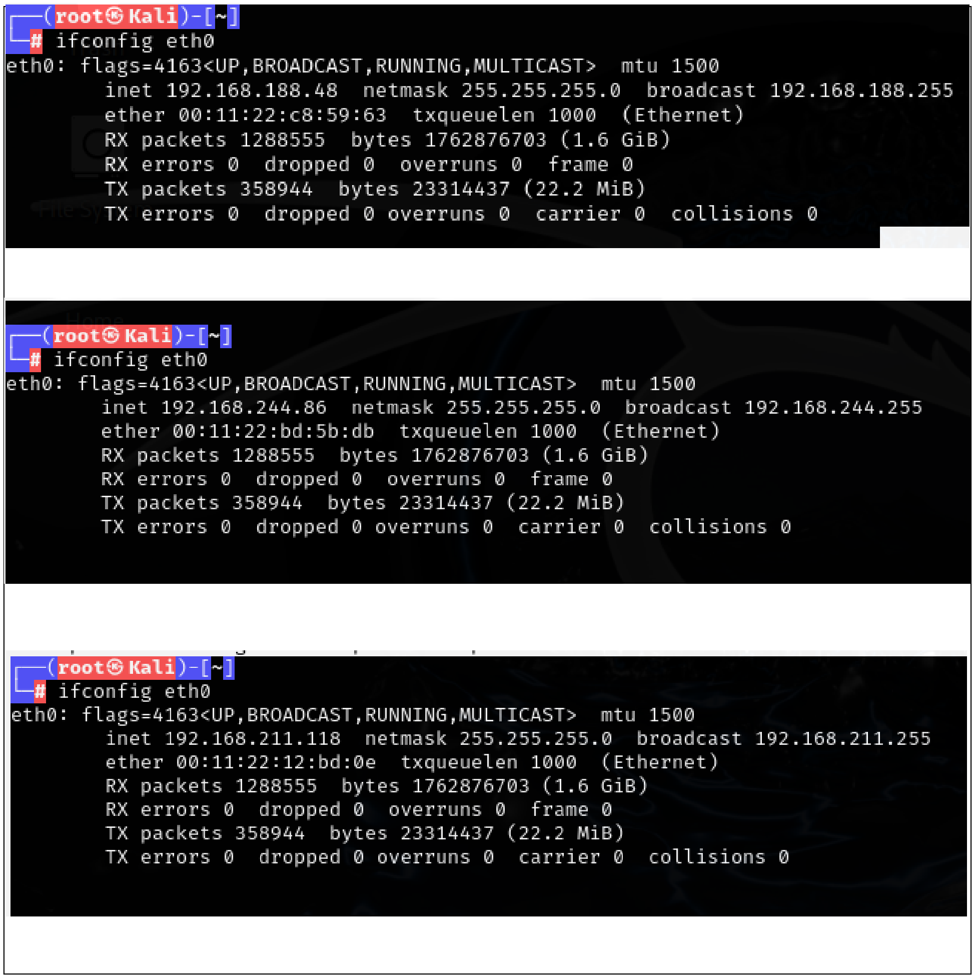
Results

5.1 Address Changes

The script successfully changed addresses every 10 seconds with output verification.

5.2 Tor Testing

Tor IP verification confirmed anonymous routing through the network.



```
┌──(root㉿Kali)-[~]
└─# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.188.48  netmask 255.255.255.0  broadcast 192.168.188.255
        ether 00:11:22:c8:59:63  txqueuelen 1000  (Ethernet)
        RX packets 1288555  bytes 1762876703 (1.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 358944  bytes 23314437 (22.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
┌──(root㉿Kali)-[~]
└─# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.244.86  netmask 255.255.255.0  broadcast 192.168.244.255
        ether 00:11:22:bd:5b:db  txqueuelen 1000  (Ethernet)
        RX packets 1288555  bytes 1762876703 (1.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 358944  bytes 23314437 (22.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
┌──(root㉿Kali)-[~]
└─# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.211.118  netmask 255.255.255.0  broadcast 192.168.211.255
        ether 00:11:22:12:bd:0e  txqueuelen 1000  (Ethernet)
        RX packets 1288555  bytes 1762876703 (1.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 358944  bytes 23314437 (22.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 5.1: Sample terminal output showing address changes

Chapter 6

Conclusion

This project successfully demonstrated automated IP/MAC spoofing with Tor integration, providing an effective solution for network anonymity. The implementation shows potential for privacy protection in security-sensitive operations.

Chapter 7

Future Work

- Network condition-based automatic switching
- Multiple proxy integration
- Enhanced monitoring capabilities

Bibliography

[1] Tor Project. `https://www.torproject.org/`

[2] macchanger Documentation. `https://man7.org/linux/man-pages/man1/macchanger.1.html`

[3] Python Subprocess Module. `https://docs.python.org/3/library/subprocess.html`