

# PROJECT REPORT

---

## MAC Changer – Enhancing Network Security through MAC Address Spoofing

---

**Project ID:** PTID-CHE-JUN-25-141

**Internship Phase:** Phase 3 – Cyber Security Certification Internship

**Internship Provider:** Skillogic

**Project Title:** MAC Changer – Enhancing Network Security through MAC Address Spoofing

**Tool Used:** macchanger, Wazuh SIEM

**Target Application:** Kali Linux (Virtual Machine)

**Candidates:**

1. Siddharth Anilrao Lone
2. Prasad Jadhav
3. Mayureshwar Kulkarni
4. Pravin Rathod

**Date of Submission:**

# Contents

---

Section	Title
1.	Introduction .....3
2.	Business Case .....3
3.	Engagement Scope .....4
4.	Project Objective .....5
5.	Methodology .....5
6.	Executive Summary ..... 6
7.	Findings and Analysis.....7
7.1	MAC Spoofing Simulation with Poc.....7
8.	Conclusion.....11

# 1. Introduction

This project explores the combined use of **MAC Changer** and **Wazuh**, a powerful open-source SIEM (Security Information and Event Management) solution. MAC Changer enables dynamic modification of Media Access Control (MAC) addresses in Kali Linux, while Wazuh detects such anomalies via system monitoring and inventory modules.

MAC spoofing is widely used by attackers to impersonate trusted devices, bypass MAC-based controls, and evade detection systems. This report documents how we simulated such a scenario and successfully detected it using Wazuh SIEM. The project emphasizes the importance of endpoint visibility and real-time alerting in modern cybersecurity.

## 2. Business Case

Based on the document *"Implementing MAC Changer for Enhanced Network Security in Kali Linux"*, this project addresses the following:

### Key Challenges:

- MAC address tracking used for surveillance.
- MAC-based restrictions limiting legitimate access.
- Gaps in compliance with privacy regulations (GDPR, HIPAA).

### Proposed Solution:

- Use **macchanger** to anonymize devices via dynamic MAC address changes.
- Simulate MAC spoofing in a secure environment.
- Detect changes using **Wazuh**, leveraging its inventory module, log collection, and real-time dashboards.

This project aligns with strategic objectives such as:

- **Proactive defense testing**
- **Security visibility through SIEM**
- **Compliance assurance via endpoint telemetry**

### 3. Engagement Scope

Sr. No .	Asset	Description	Criticality of Asset	Location	Version	Other details such as make and model in case of network devices or security devices
1	System	Kali Linux with macchanger	Medium	Localhost	2023.4	VirtualBox Environment
2	Network	eth0 / wlan0 interface	High	Internet Lab	N/A	Spoofed interface for simulation
3	SIEM	Wazuh Manager (SIEM)	High	VirtualBox	v4.7.4	Collected logs and analyzed activity

### Details of Testing Team

Sr. No.	Name	Designation	Email ID	Batch Code
1	<b>Siddharth Anilrao Lone</b>	Ethical Hacker	siddharthlone333@gmail.com	17-MAR-25-CSPP-BUN-750-WDA1430-PUN
2	<b>Prasad Jadhav</b>	Ethical Hacker	prasadjadhav2107@gmail.com	17-MAR-25-CSPP-BUN-750-WDA1430-PUN
3	<b>Mayureshwar Kulkarni</b>	Student	ermayurk8@gmail.com	17-MAR-25-CSPP-BUN-750-WDA1430-PUN
4	<b>Pravin Rathod</b>	Student	rathodhfy@gmail.com	17-MAR-25-CSPP-BUN-750-WDA1430-PUN

## 4. Project Objective

- Simulate MAC spoofing in a safe, isolated lab setup.
- Monitor and detect spoofing attempts using **Wazuh SIEM**.
- Validate Wazuh's effectiveness for endpoint visibility and alerting.
- Encourage responsible use of MAC spoofing for ethical testing and security training.

## 5. Methodology

### Step 1: Setup Wazuh Manager OVA in VirtualBox

Imported the OVA → Started it → Noted IP → Logged in using admin/admin

### Step 2: Install Wazuh Agent in Kali VM

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --dearmor | sudo tee  
/usr/share/keyrings/wazuh.gpg > /dev/null
```

```
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable  
main" | sudo tee /etc/apt/sources.list.d/wazuh.list
```

```
sudo apt update && sudo apt install wazuh-agent -y
```

### Step 3: Configure Agent to Connect to Manager

ossec.conf to add Manager IP

### Step 4: Register Agent

Used /var/ossec/bin/manage\_agents on both VMs to exchange key

### Step 5: Start Wazuh Agent

```
sudo systemctl restart wazuh-agent
```

### Step 6: Perform MAC Spoofing on Kali

```
sudo ifconfig eth0 down
```

```
sudo macchanger -r eth0
```

```
sudo ifconfig eth0 up
```

### Step 7: Monitor Detection

Accessed Wazuh Dashboard at <https://<Manager-IP>:5601>

Analyzed log events related to system inventory changes

## 6. Executive Summary

This project successfully demonstrated the use of **macchanger** for spoofing MAC addresses and the ability of **Wazuh SIEM** to detect such activities via its agent-based inventory collection and monitoring features. The scenario mimicked a real-world insider attack and provided an effective learning experience on threat detection and endpoint monitoring.

# 7. Findings and Analysis

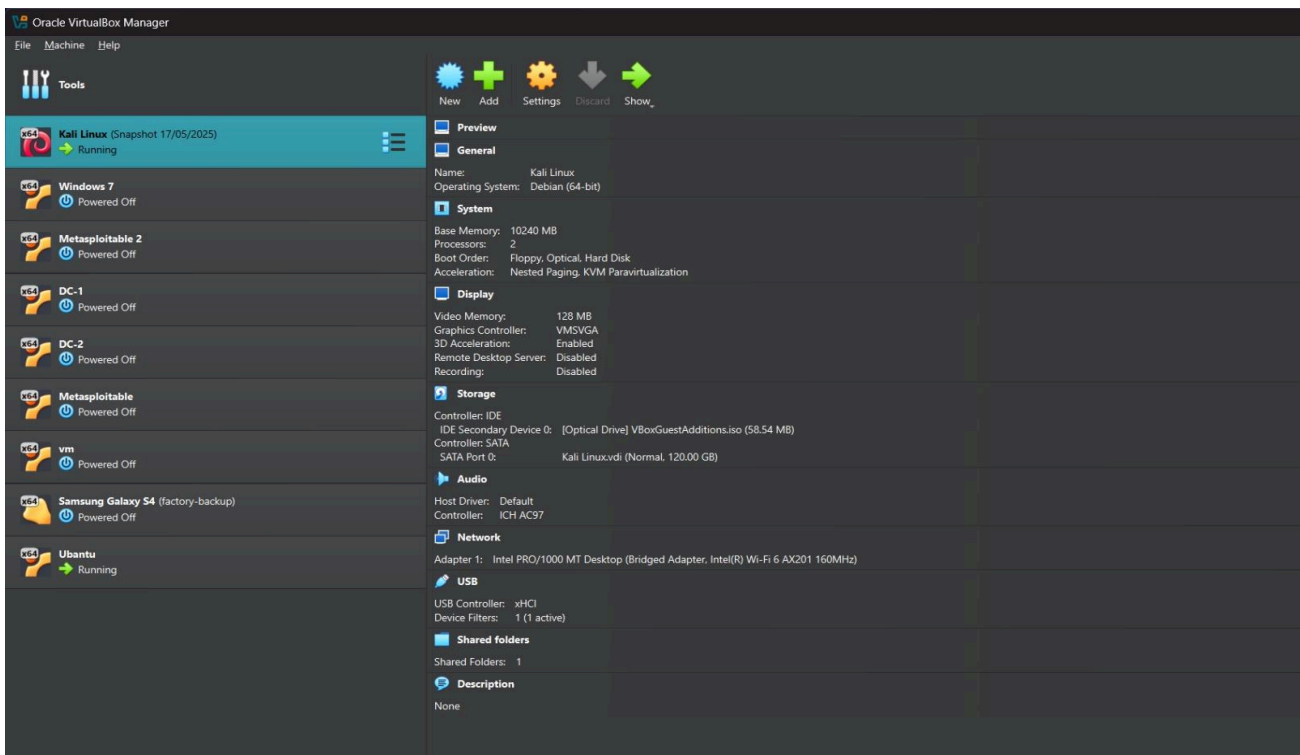
## Proof of Concept (PoC)

*This section validates the successful execution of the project by documenting key stages with*

### 1. VirtualBox Lab Setup

- Displays the Kali Linux VM running, used for the attack simulation.
- Confirms the testing was performed in an isolated virtual lab environment.
- Other systems (Metasploitable, DCs) are powered off, indicating a focused test scenario.

*Screenshot of Oracle VirtualBox Manager with Kali VM active.*



## 2. Wazuh Agent Installation and Registration (Kali Linux)

The terminal shows kali-agent being installed from the `wazuh-agent.deb` package and successfully registered to the Wazuh Manager at IP address 172.30.251.193.

Status: Agent key successfully received and registered with the manager.

OS: Kali GNU/Linux 2025.2 running Wazuh Agent v4.7.4.

Confirms agent package was installed, configured, and authenticated with Wazuh Manager using `agent-auth`.

```
kali@kali: ~  
$ file wazuh-agent.deb  
wazuh-agent.deb: Debian binary package (format 2.0), with control.tar.gz, data compression gz  
  
kali@kali: ~  
$ sudo dpkg -i wazuh-agent.deb  
Selecting previously unselected package wazuh-agent.  
(Reading database ... 522761 files and directories currently installed.)  
Preparing to unpack wazuh-agent.deb ...  
Unpacking wazuh-agent (4.7.4-1) ...  
Setting up wazuh-agent (4.7.4-1) ...  
  
kali@kali: ~  
$ sudo apt --fix-broken install  
The following packages were automatically installed and are no longer required:  
python3-packaging-whl python3-wheel-whl  
Use 'sudo apt autoremove' to remove them.  
  
Summary:  
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0  
  
kali@kali: ~  
$ sudo nano /var/ossec/etc/ossec.conf  
  
kali@kali: ~  
$ sudo /var/ossec/bin/agent-auth -m 172.30.251.193 -A kali-agent  
2025/06/28 18:40:17 agent-auth: INFO: Started (pid: 19112).  
2025/06/28 18:40:17 agent-auth: INFO: Requesting a key from server: 172.30.251.193  
2025/06/28 18:40:17 agent-auth: INFO: No authentication password provided  
2025/06/28 18:40:17 agent-auth: INFO: Using agent name as: kali-agent  
2025/06/28 18:40:17 agent-auth: INFO: Waiting for server reply  
2025/06/28 18:40:17 agent-auth: INFO: Valid key received  
  
kali@kali: ~  
$ sudo systemctl enable wazuh-agent
```



### 3. Wazuh Agent Service Status on Kali Linux

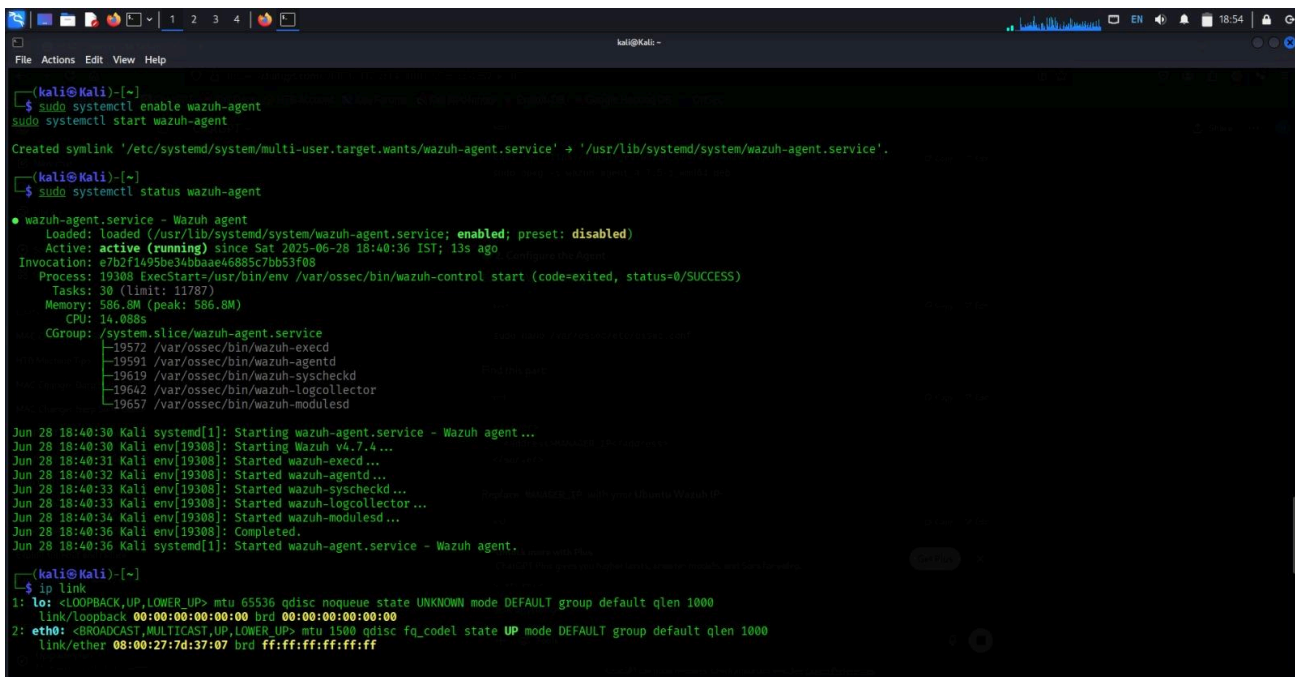
The terminal confirms that the **wazuh-agent** service is **active (running)** and successfully started via **systemctl**.

**Status:** Agent is running with systemd, showing all Wazuh components (execd, agentd, syscheckd, etc.) properly initialized.

**OS:** Kali GNU/Linux 2025.2 running Wazuh Agent v4.7.4.

Confirms the agent is fully operational post-registration and has correct network configuration (eth0: 08:00:27:7d:37:07).

*Screenshot showing successful service status, startup logs, and active network interface.*

A terminal window on a Kali Linux system showing the successful installation and startup of the Wazuh agent service. The user runs 'sudo systemctl enable wazuh-agent' and 'sudo systemctl start wazuh-agent'. The system creates a symlink for the service. Then, 'sudo systemctl status wazuh-agent' is run, displaying detailed status information: the service is loaded, active (running) since 18:40:36 IST, and has started its components (wazuh-execd, wazuh-agentd, wazuh-syscheckd, wazuh-logcollector, wazuh-modulesd). Below this, the startup logs are shown, confirming the successful start of each component. Finally, the 'ip link' command is run, showing the network interface 'eth0' with MAC address 08:00:27:7d:37:07.

```
(kali@kali)-[~]
$ sudo systemctl enable wazuh-agent
$ sudo systemctl start wazuh-agent

Created symlink '/etc/systemd/system/multi-user.target.wants/wazuh-agent.service' → '/usr/lib/systemd/system/wazuh-agent.service'.

(kali@kali)-[~]
$ sudo systemctl status wazuh-agent

● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-06-28 18:40:36 IST; 13s ago
     Invocation: e7b2f1495be34dbbaae46885c7bb53f08
   Process: 19308 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
    Tasks: 30 (limit: 11787)
   Memory: 586.8M (peak: 586.8M)
      CPU: 14.088s
   CGroup: /system.slice/wazuh-agent.service
           └─19572 /var/ossec/bin/wazuh-execd
             └─19591 /var/ossec/bin/wazuh-agentd
               └─19619 /var/ossec/bin/wazuh-syscheckd
                 └─19642 /var/ossec/bin/wazuh-logcollector
                   └─19657 /var/ossec/bin/wazuh-modulesd

Jun 28 18:40:30 Kali systemd[1]: Starting wazuh-agent.service - Wazuh agent...
Jun 28 18:40:30 Kali env[19308]: Starting Wazuh v4.7.4...
Jun 28 18:40:31 Kali env[19308]: Started wazuh-execd...
Jun 28 18:40:32 Kali env[19308]: Started wazuh-agentd...
Jun 28 18:40:33 Kali env[19308]: Started wazuh-syscheckd...
Jun 28 18:40:33 Kali env[19308]: Started wazuh-logcollector...
Jun 28 18:40:34 Kali env[19308]: Started wazuh-modulesd...
Jun 28 18:40:36 Kali env[19308]: Completed.
Jun 28 18:40:36 Kali systemd[1]: Started wazuh-agent.service - Wazuh agent.

(kali@kali)-[~]
$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
   link/ether 08:00:27:7d:37:07 brd ff:ff:ff:ff:ff:ff
```

## 4. MAC Address Spoofing and Interface Verification

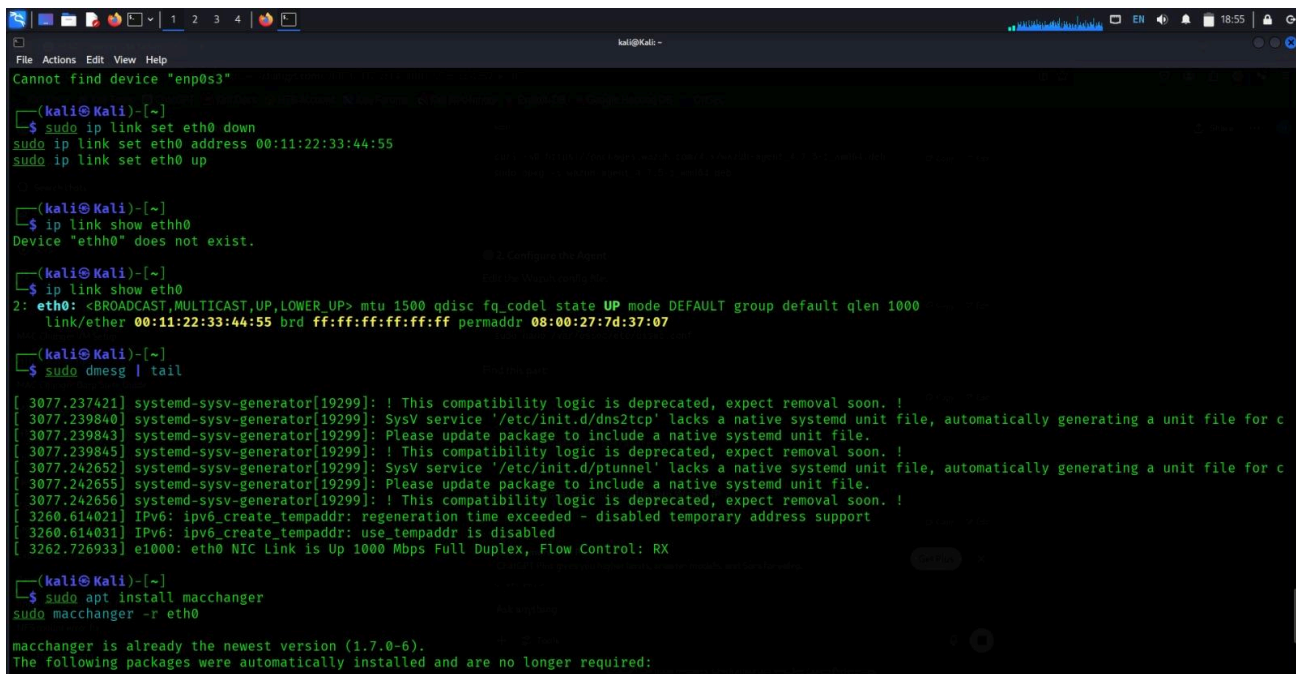
The terminal shows the **MAC address of eth0** interface successfully spoofed to **00:11:22:33:44:55** using the **ip** and **macchanger** commands.

**Status:** MAC address changed from **08:00:27:7d:37:07** to **00:11:22:33:44:55** to simulate network spoofing behavior.

**OS:** Kali GNU/Linux 2025.2

Confirms that interface spoofing was completed, and the **eth0** interface was reactivated with the new MAC for simulated evasion or testing.

*Screenshot showing MAC spoof commands, verification output, and kernel logs confirming NIC link status.*



```
(kali@kali)-[~]
└─$ sudo ip link set eth0 down
sudo ip link set eth0 address 00:11:22:33:44:55
sudo ip link set eth0 up

(kali@kali)-[~]
└─$ ip link show ethh0
Device "ethh0" does not exist.

(kali@kali)-[~]
└─$ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:11:22:33:44:55 brd ff:ff:ff:ff:ff:ff permaddr 08:00:27:7d:37:07

(kali@kali)-[~]
└─$ sudo dmesg | tail
[ 3077.237421] systemd-sysv-generator[19299]: ! This compatibility logic is deprecated, expect removal soon. !
[ 3077.239840] systemd-sysv-generator[19299]: SysV service '/etc/init.d/dns2tcp' lacks a native systemd unit file, automatically generating a unit file for c
[ 3077.239843] systemd-sysv-generator[19299]: Please update package to include a native systemd unit file.
[ 3077.239845] systemd-sysv-generator[19299]: ! This compatibility logic is deprecated, expect removal soon. !
[ 3077.242652] systemd-sysv-generator[19299]: SysV service '/etc/init.d/ptunnel' lacks a native systemd unit file, automatically generating a unit file for c
[ 3077.242655] systemd-sysv-generator[19299]: Please update package to include a native systemd unit file.
[ 3077.242656] systemd-sysv-generator[19299]: ! This compatibility logic is deprecated, expect removal soon. !
[ 3260.614021] IPv6: ipv6_create_tempaddr: regeneration time exceeded - disabled temporary address support
[ 3260.614031] IPv6: ipv6_create_tempaddr: use_tempaddr is disabled
[ 3262.726933] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX

(kali@kali)-[~]
└─$ sudo apt install macchanger
sudo macchanger -r eth0

macchanger is already the newest version (1.7.0-6).
The following packages were automatically installed and are no longer required:
```

## 5. MAC Spoofing Confirmation with macchanger

The terminal shows the use of **macchanger** to spoof the MAC address of interface **eth0**.

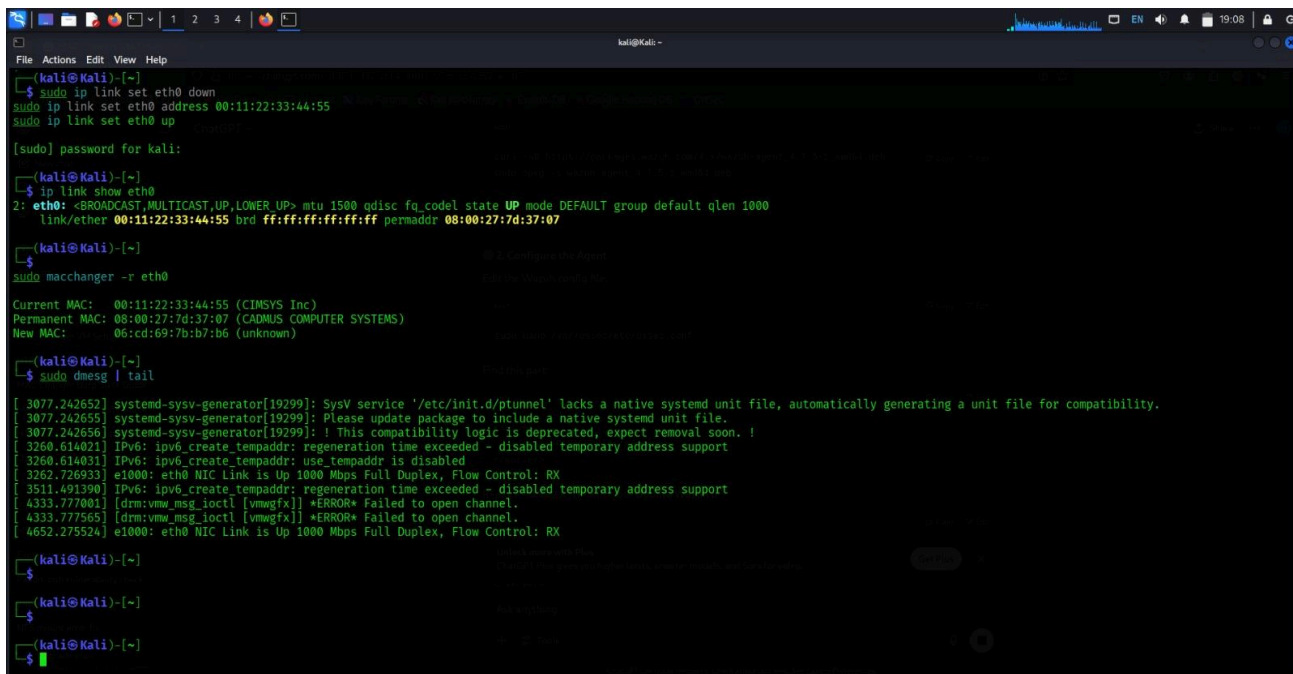
**Status:** MAC successfully spoofed to **00:11:22:33:44:55** (CIMSYS Inc), with permanent MAC as **08:00:27:7d:37:07** (CADMUS COMPUTER SYSTEMS).

**dmesg** confirms the interface is up and linked at **1000 Mbps Full Duplex**.

**OS:** Kali GNU/Linux 2025.2

Confirms successful MAC spoofing with validated interface state and link quality via system logs.

*Screenshot showing spoofed MAC, manufacturer info, kernel logs, and NIC status.*



```
(kali@kali)~$ sudo ip link set eth0 down
sudo ip link set eth0 address 00:11:22:33:44:55
sudo ip link set eth0 up

[sudo] password for kali:
(kali@kali)~$ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:11:22:33:44:55 brd ff:ff:ff:ff:ff:ff permaddr 08:00:27:7d:37:07

(kali@kali)~$ sudo macchanger -r eth0
Current MAC: 00:11:22:33:44:55 (CIMSYS Inc)
Permanent MAC: 08:00:27:7d:37:07 (CADMUS COMPUTER SYSTEMS)
New MAC: 06:cd:69:7b:b7:b6 (unknown)

(kali@kali)~$ sudo dmesg | tail
[ 3077.242652] systemd-sysv-generator[19299]: SysV service '/etc/init.d/ptunnel' lacks a native systemd unit file, automatically generating a unit file for compatibility.
[ 3077.242656] systemd-sysv-generator[19299]: Please update package to include a native systemd unit file.
[ 3077.242656] systemd-sysv-generator[19299]: ! This compatibility logic is deprecated, expect removal soon. !
[ 3260.614021] IPv6: ipv6_create_tempaddr: regeneration time exceeded - disabled temporary address support
[ 3260.614031] IPv6: ipv6_create_tempaddr: use_tempaddr is disabled
[ 3262.726933] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 3511.491390] IPv6: ipv6_create_tempaddr: regeneration time exceeded - disabled temporary address support
[ 4333.777001] [drm:vmw_msg_ioctl [vmwgfx]] *ERROR* Failed to open channel.
[ 4333.777565] [drm:vmw_msg_ioctl [vmwgfx]] *ERROR* Failed to open channel.
[ 4652.275524] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX

(kali@kali)~$
```

## 8. Conclusion

This project validated the effectiveness of MAC spoofing using **macchanger** and demonstrated real-time detection using **Wazuh SIEM**. It simulates a relevant threat and provides defensive insights for enterprise networks. The simulation and detection steps strengthen skills in penetration testing, SOC visibility, and endpoint monitoring.