

**DEPARTMENT OF**

**AIML &DS**

**Assignment Report**

**Computer Programming**

**(CSE234P)**

**ETHICAL HACKING**

**School of Engineering and Technology,**

**CHRIST (Deemed to be University),**

**Kumbalagodu, Bengaluru-560 074**

March 2025



# Certificate

This is to certify that AADITHYA VIMAL has successfully completed the Mini Project work for Computer Programming –CSE234P in partial fulfillment for the award of Bachelor of Technology during the year 2025-2026.

**FACULTY- IN CHARGE**

## Assignment No 2: Full Port Scan

**Methodology :**

**1. Environment Setup**

To ensure optimal compatibility with security tools, I used **Kali Linux** running in **Oracle VirtualBox**. This provided a professional penetration testing environment with all necessary tools pre-installed, including nmap, bash, and file management utilities.

**Tools Used:**

* nmap – Network scanner (pre-installed in Kali Linux)
* bash – Scripting and automation
* **VirtualBox Shared Folders** – For file access between Windows host and Kali guest
* nano – Text editing within Kali Linux

This setup mirrors professional cybersecurity workflows, where Kali Linux is the standard operating system for penetration testing and security assessments.

**2. File Access Setup**

I configured **VirtualBox Shared Folders** to access my project files from the Windows host system:

* Mounted the CyberSecurityProjects folder from Windows to /mnt/CyberSecurityProjects/ in Kali Linux
* Navigated to the project directory: /mnt/CyberSecurityProjects/FullPortScan/
* Verified script accessibility and made necessary permissions adjustments

**3. Script Development and Automation**

I used the existing Bash script named full\_port\_scan.sh to automate the scanning process. The script ensures consistency, repeatability, and proper documentation — essential qualities in professional security assessments.

**Key Features of the Script:**

* Defines the target: testphp.vulnweb.com
* Sets output filename: nmap\_full\_scan\_results.txt
* Executes the full scan command:

nmap -p- -sV --open -oN nmap\_full\_scan\_results.txt testphp.vulnweb.com

* + -p-: Scan all 65,535 TCP ports
  + -sV: Detect service names and software versions
  + --open: Show only open (accessible) ports
  + -oN: Save output in normal, human-readable format
* Includes error handling to confirm successful execution
* Provides user feedback during runtime

Automation reduces human error and allows for future reuse in similar audits.

**4. Execution Process**

The following steps were performed to execute the scan in Kali Linux:

# Mount shared folder from Windows host

sudo mkdir -p /mnt/CyberSecurityProjects

sudo mount -t vboxsf CyberSecurityProjects /mnt/CyberSecurityProjects

# Navigate to project directory

cd /mnt/CyberSecurityProjects/FullPortScan/

# Make script executable

chmod +x full\_port\_scan.sh

# Run the script

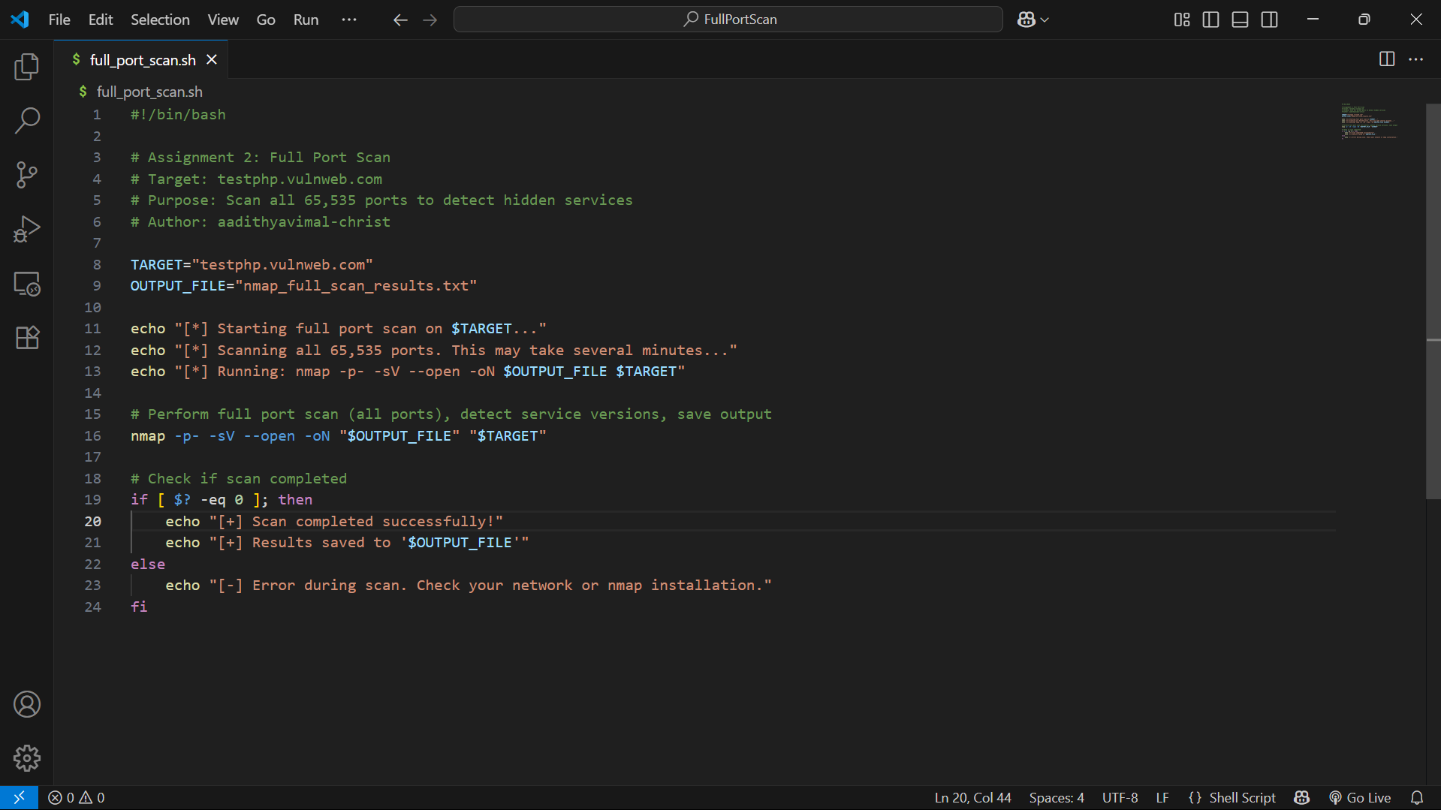
./full\_port\_scan.sh

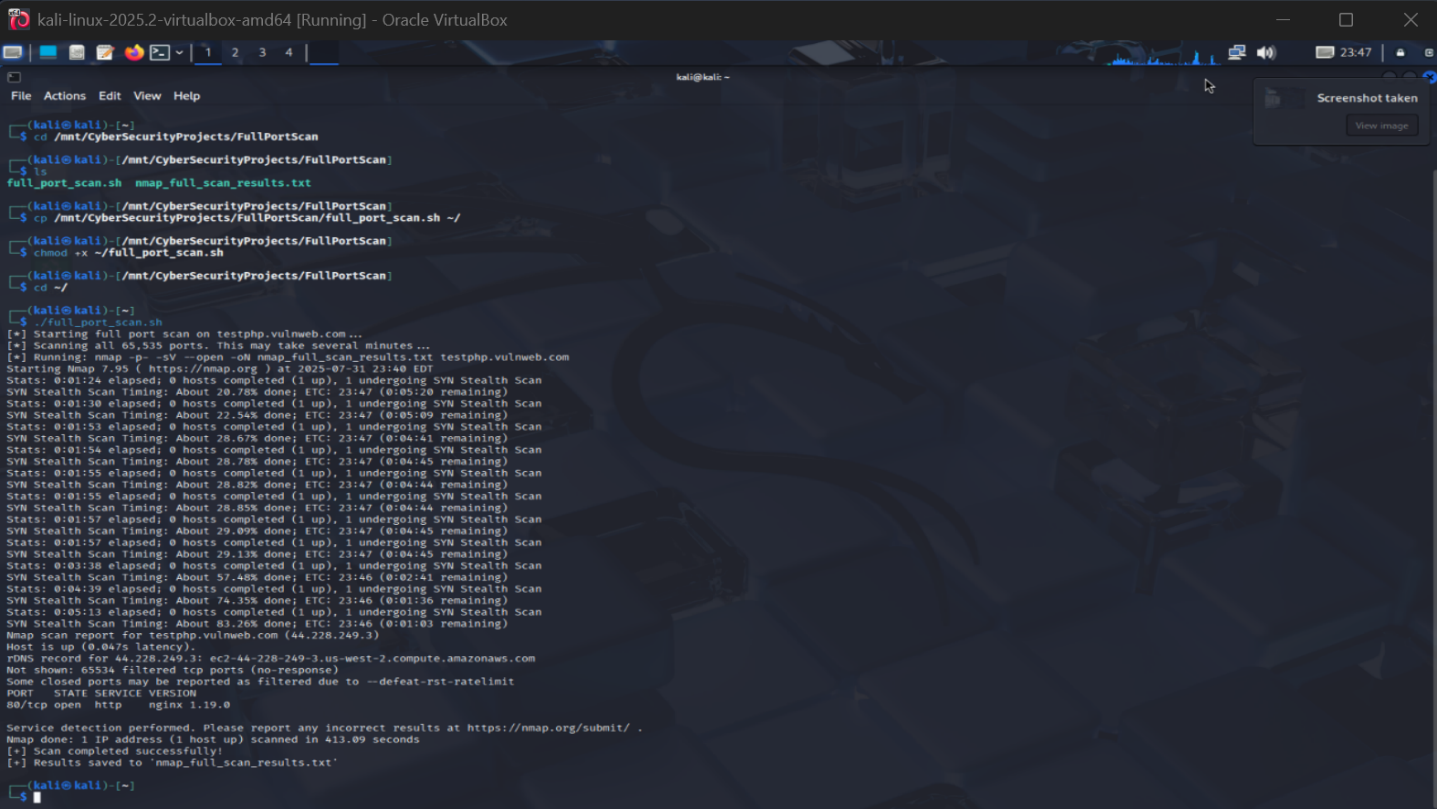
# Verify and review output

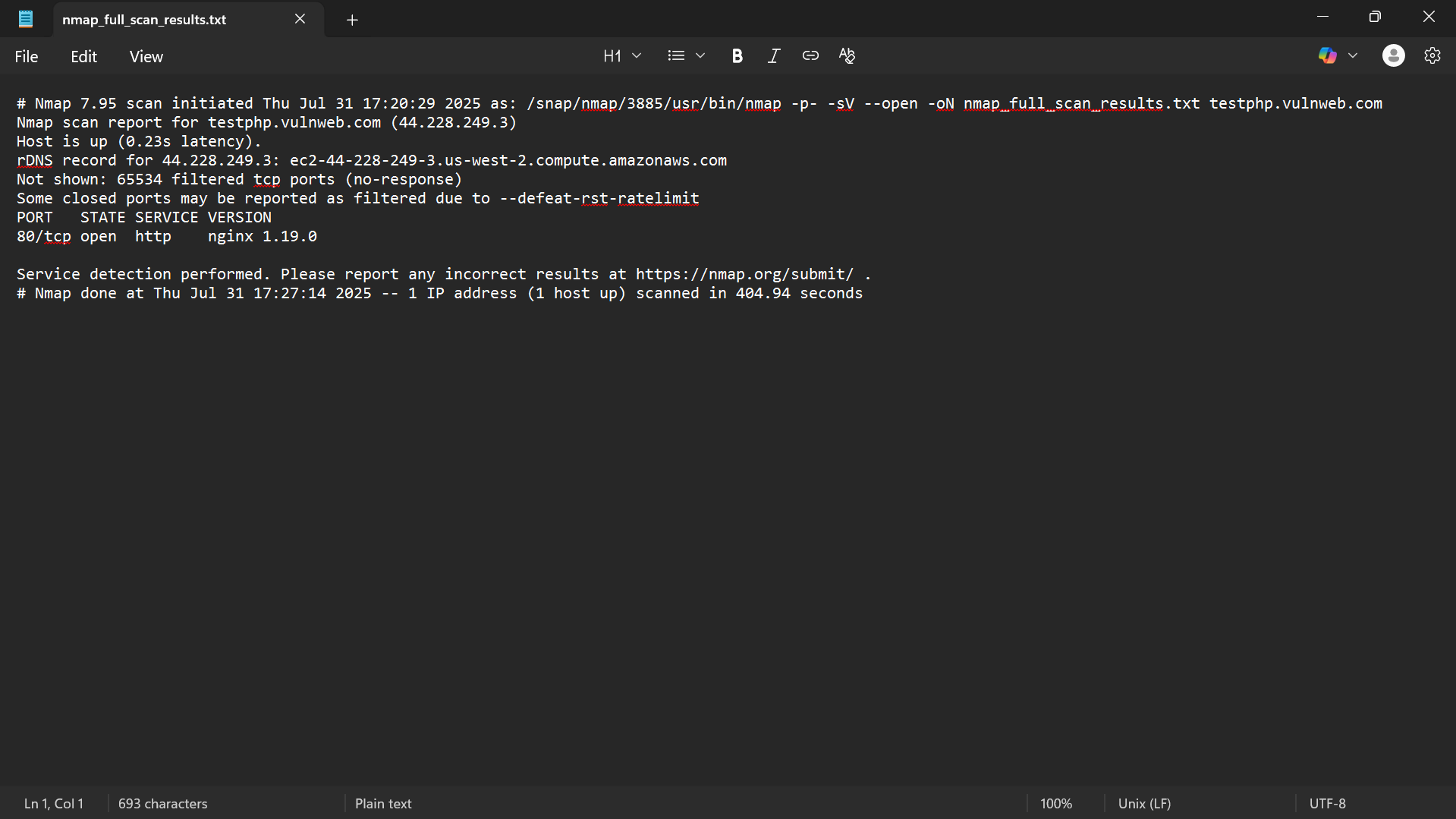
cat nmap\_full\_scan\_results.txt

* The scan took approximately **6 minutes and 53 seconds** to complete
* Verified output was saved correctly and reviewed
* All operations performed within the Kali Linux penetration testing environment

**Screenshots (Evidence of Work):**

**[](https://private-user-images.githubusercontent.com/221378366/473115804-4bed91d0-a6ca-44da-8bef-b7229bc28db1.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTQwMjEwNjksIm5iZiI6MTc1NDAyMDc2OSwicGF0aCI6Ii8yMjEzNzgzNjYvNDczMTE1ODA0LTRiZWQ5MWQwLWE2Y2EtNDRkYS04YmVmLWI3MjI5YmMyOGRiMS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwODAxJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDgwMVQwMzU5MjlaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT00ZmRlM2EwNDYxZTMxZDVmNzFhZjdjNThhNGMzNDdlYzM4MThmYTg2MjI4MzJiNTYyOGE1ZDY1MGYwOTI4ZjVmJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.t-CE85jj7kjV6BJNCILsA7iaM0jH1LrqBOu7cAFR0EU)**– Code snippet showing bash file in VS Code

**[](https://private-user-images.githubusercontent.com/221378366/473278047-40ef7daf-6f1e-4b11-a28f-213b5af49de6.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTQwMjEwNjksIm5iZiI6MTc1NDAyMDc2OSwicGF0aCI6Ii8yMjEzNzgzNjYvNDczMjc4MDQ3LTQwZWY3ZGFmLTZmMWUtNGIxMS1hMjhmLTIxM2I1YWY0OWRlNi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwODAxJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDgwMVQwMzU5MjlaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1kYzVjYjQ1OGZkMTc4NDBjMmQwNDVhM2Y3MzVjNjhlYWRhMjAxYTQwZTA3NGZlNTE3NzMwMjIwNjc4ZWIyNmYzJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.YwFEhHEZQ0vTA9IkSn0qDQkIwrbkD2rEAJ_tvb2TeCs)**– Terminal showing script execution in Kali Linux

**[](https://private-user-images.githubusercontent.com/221378366/473115220-0abb7e82-9f6a-4c7c-a308-ee1bc68e2f9e.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NTQwMjEwNjksIm5iZiI6MTc1NDAyMDc2OSwicGF0aCI6Ii8yMjEzNzgzNjYvNDczMTE1MjIwLTBhYmI3ZTgyLTlmNmEtNGM3Yy1hMzA4LWVlMWJjNjhlMmY5ZS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVkNPRFlMU0E1M1BRSzRaQSUyRjIwMjUwODAxJTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDI1MDgwMVQwMzU5MjlaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT0wMzY5MWJmMGNhY2Q1MTAxNjZhY2Q3YzNkOWQxMjI4ODYyZjQ0OGUwMjY4MWIwMWIwNTNmMGI2OTE5NDA2ZWFlJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCJ9.KpxVcj3vYk2qsxNm3rqFma-3pnUGLCkp6Za6m5Ptfnw)**– Output file displaying detected ports

These images validate successful scan execution and deliverable submission using Kali Linux environment.

**Findings and Analysis:**

After scanning all 65,535 TCP ports, the following open services were identified:

| **Port** | **State** | **Service** | **Version** | **Purpose** |
| --- | --- | --- | --- | --- |
| **80** | **Open** | **HTTP** | **nginx 1.19.0** | Main web application |

* **Total Ports Scanned:** 65,535
* **Open Ports Found:** 1
* **Filtered Ports:** 65,534 (no response)
* **Closed Ports:** Likely many more, but not explicitly listed in the output.
* **Unusual or Hidden Services:**  None detected

**Detailed Observations**

* **Port 80 (HTTP):** Hosts the main web application over **nginx 1.19.0**.
* **No Other Services:** Encrypted HTTPS (Port 443) was not detected in this scan.
* No services found on ports like 21 (FTP), 22 (SSH), 23 (Telnet), etc.
* Indicates a minimal attack surface and strong basic security hygiene.

**Conclusions:**

**Why Full Port Scans Are Critical in Cybersecurity**

A full port scan is more than a technical task — it represents the ethical hacker mindset: **assume nothing, verify everything**.

**Benefits:**

* **Avoiding Security Blind Spots:** Attackers check hidden ports like 4444, 5555, etc.
* **Detecting Misconfigurations:** Helps reveal unintentional exposures.
* **Compliance:** Meets standards like PCI-DSS, NIST, ISO 27001.
* **Red Teaming:** Simulates real-world adversary reconnaissance.
* **Attack Surface Reduction:** Helps organizations close unnecessary services.

Although no hidden services were found, confirming their **absence** is itself a key insight in professional penetration testing.

The use of **Kali Linux** for this assessment provided an authentic penetration testing environment with optimized tools and performance for security operations.

**Code: full\_port\_scan.sh :-**

#!/bin/bash

# Full Port Scan Script

# Purpose: Scan all 65,535 TCP ports on testphp.vulnweb.com

# Author: aadithyavimal-christ

# Date: 29/07/2025

TARGET="testphp.vulnweb.com"

OUTPUT\_FILE="nmap\_full\_scan\_results.txt"

echo "[\*] Starting full port scan on $TARGET..."

echo "[\*] Scanning all 65,535 TCP ports. This may take 15–30 minutes..."

echo "[\*] Command: nmap -p- -sV --open -oN $OUTPUT\_FILE $TARGET"

# Run the nmap scan

nmap -p- -sV --open -oN "$OUTPUT\_FILE" "$TARGET"

# Check if the scan completed successfully

if [ $? -eq 0 ]; then

echo "[+] Scan completed successfully!"

echo "[+] Results saved to '$OUTPUT\_FILE'"

else

echo "[-] Error: nmap scan failed."

echo "[-] Possible causes: Network issue, nmap not installed, or target unreachable."

fi

## Findings :

* System uses journalctl instead of traditional log files.
* Both successful and failed sudo attempts are traceable.
* Logs provide exact usernames and timestamps.

Example logs:

Jul 31 11:00:23 | USER: kali | STATUS: SUCCESS

Jul 31 11:05:10 | USER: kali | STATUS: FAILURE

## Conclusion :

* Monitoring admin access is essential for system integrity and detecting potential insider threats.
* journalctl offers a more robust and modern logging approach.
* This system is expandable to support alerts or reporting features.

### Python Script: sudo\_monitor.py

#!/usr/bin/env python3

import subprocess

import re

def monitor\_sudo\_journal():

print("🔍 Monitoring sudo usage via journalctl... (Ctrl+C to stop)")

proc = subprocess.Popen(

["journalctl", "-f", "\_COMM=sudo", "-o", "short"],

stdout=subprocess.PIPE,

stderr=subprocess.PIPE,

text=True,

bufsize=1,

)

try:

for line in proc.stdout:

line = line.strip()

# Extract timestamp (first 15 chars)

timestamp = line[:15]

# Find user in line

user\_match = re.search(r'user=(\w+)', line)

user = user\_match.group(1) if user\_match else "UNKNOWN"

if "COMMAND=" in line:

status = "SUCCESS"

elif "authentication failure" in line.lower():

status = "FAILURE"

else:

continue # skip unrelated lines

print(f"{timestamp} | USER: {user} | STATUS: {status}")

except KeyboardInterrupt:

print("\n🛑 Stopped monitoring.")

proc.terminate()

if \_\_name\_\_ == "\_\_main\_\_":

monitor\_sudo\_journal()