DEPARTMENT OF AIML & DS
Assignment No 2: Full Port Scan
Computer Programming (CSE234P)
ETHICAL HACKING

Student: AADITHYA VIMAL
Course: B.Tech (AIML & DS)

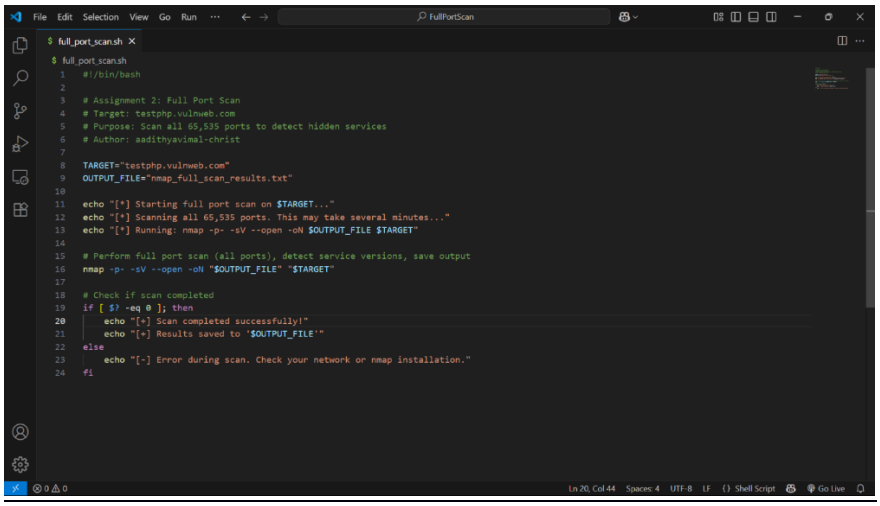
Objective: Perform a full TCP port scan on testphp.vulnweb.com using automated scripting in Kali Linux.

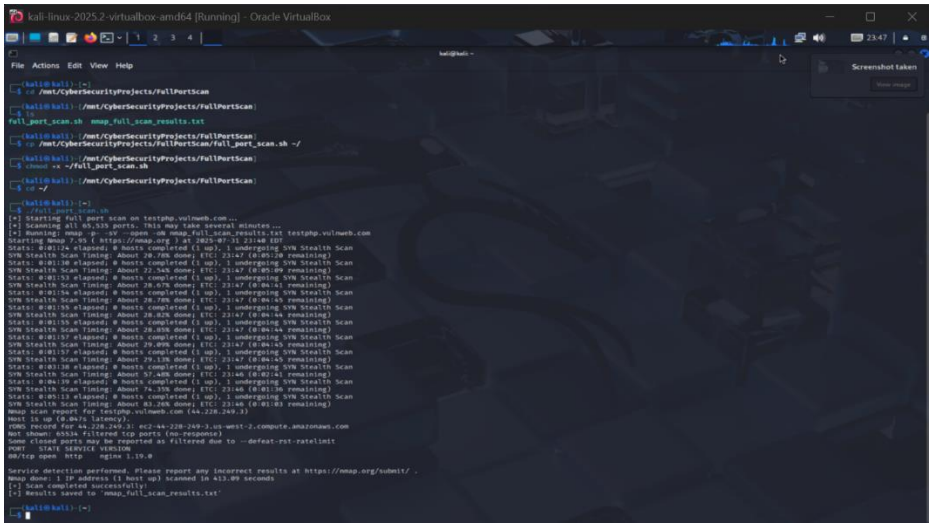
Methodology:
Used Kali Linux (VirtualBox) with nmap, bash, and shared folders for file access. Developed and executed a Bash script (full_port_scan.sh) to automate scanning all 65,535 TCP ports with service detection.

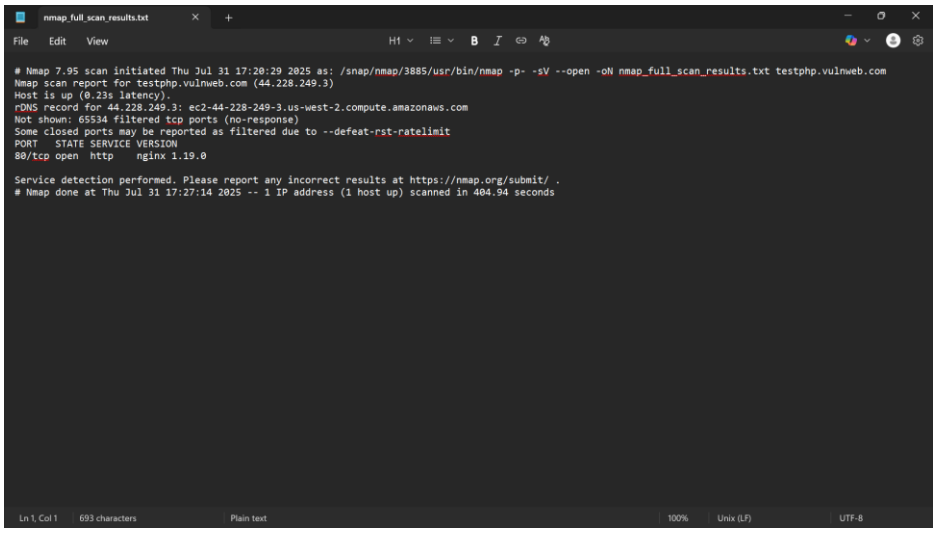
Scan Command:
nmap -p- -sV --open -oN nmap_full_scan_results.txt testphp.vulnweb.com

- -p-: Full port range
- -sV: Service/version detection
- --open: Show only open ports
- -oN: Save readable output

Screenshots:







Execution Steps:

1. Mounted shared folder:
   sudo mount -t vboxsf CyberSecurityProjects /mnt/CyberSecurityProjects
2. Navigated to project: cd /mnt/CyberSecurityProjects/FullPortScan/
3. Made script executable: chmod +x full_port_scan.sh
4. Ran script: ./full_port_scan.sh

Findings:

PORT	SERVICE	VERSION	STATUS
80	HTTP	nginx 1.19.0	Open

- Total Ports Scanned: 65,535
- Open Ports: 1

- Filtered/Closed: 65,534 (no response or closed)
- No hidden services (e.g., SSH, FTP, Telnet) detected.
- Minimal attack surface; no HTTPS (port 443) found.

Script Highlights (full_port_scan.sh):

10

```
#!/bin/bash
TARGET="testphp.vulnweb.com"
OUTPUT_FILE="nmap_full_scan_results.txt"
echo "[*] Starting full port scan..."
nmap -p- -sV --open -oN "$OUTPUT_FILE" "$TARGET"
if [ $? -eq 0 ]; then
   echo "[+] Scan completed successfully!"
else
   echo "[-] Scan failed."
fi
```

Additional Task: Sudo Activity Monitoring (Python)
Developed sudo_monitor.py using journalctl to track sudo usage in real time:

python

```
#!/usr/bin/env python3
import subprocess, 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)
   try:
      for line in proc.stdout:
         line = line.strip()
         timestamp = line[:15]
```

```
            user = re.search(r'user=(\w+)', line)
            user = user.group(1) if user else "UNKNOWN"
            status = "SUCCESS" if "COMMAND=" in line else "FAILURE" if "authentication failure"
in line.lower() else None
            if status: print(f"{timestamp} | USER: {user} | STATUS: {status}")
    except KeyboardInterrupt:
        print("\n🛑 Stopped monitoring.")
        proc.terminate()
if __name__ == "__main__":
    monitor_sudo_journal()
```

Logs Sample:

```
Jul 31 11:00:23 | USER: kali | STATUS: SUCCESS
Jul 31 11:05:10 | USER: kali | STATUS: FAILURE
```

Conclusion:

Full port scanning ensures no hidden services are missed, critical for penetration testing and compliance (PCI-DSS, NIST). The target exposes only HTTP (port 80), indicating good security hygiene. Real-time sudo monitoring via journalctl enhances system accountability and threat detection.