**Nmap Vulnerability Assessment Report**

**Task Number:** 2.1 Nmap Scan Report

**Name:** Jai Rane

**Intern ID:** ASPL2518045

**Date:** 18-10-2025

# 1. Tools & Methodology

1. Tool Used: Nmap v7.94SVN
2. Operating System Environment: Kali Linux on Oracle VirtualBox
3. Scan Type: TCP SYN (Stealth) Scan, Service Version Detection, and OS Detection, etc

# 2. Commands used

1. nmap -sN 192.168.x.x : TCP NULL scan: sends TCP packets with no flags set to probe which ports respond (good for stealthy detection).
2. nmap -sS 192.168.x.x : TCP SYN (half-open) scan: sends SYNs and watches for SYN/ACKs to quickly find open ports without completing the handshake.
3. nmap -sV 192.168.x.x : Version detection: connects to open ports to ask services what they are and returns software names and versions.
4. nmap -sU 192.168.x.x : UDP scan: probes UDP ports to discover services that don’t use TCP (slower and noisier, but important).
5. nmap -O 192.168.x.x : OS detection: analyses responses to guess the target’s operating system and device type

# 3. Findings

1. nmap -sN 192.168.x.x

* What it does: TCP **NULL** scan — sends packets with **no flags** set (a stealthy probe).
* Expected findings: Ports often show as **open**, **closed** or **filtered**; useful to spot responses from weird/older stacks — you’ll typically see a short list of ports that replied (or nothing if filtered).
* Quick interpretation: If a port responds, it may be open; no-response usually looks **filtered** (firewall dropped it).

2. nmap -sS 192.168.x.x

* What it does: TCP **SYN** (“half-open”) scan — sends SYN and watches for SYN/ACK (fast and common).
* Expected findings: A neat table of ports with open (SYN/ACK received), closed (RST received), or filtered (no reply).
* Quick interpretation: open means a service is listening; good baseline for which services to investigate further.

3. nmap -sV 192.168.x.x

* What it does: **Version detection** — probes open ports to ask services what they are (banner/response analysis).
* Expected findings: Service names and versions next to each open port (e.g., 80/tcp open http Apache httpd 2.2.8).
* Quick interpretation: Use these version strings to map to known CVEs or decide if a service is outdated/vulnerable.

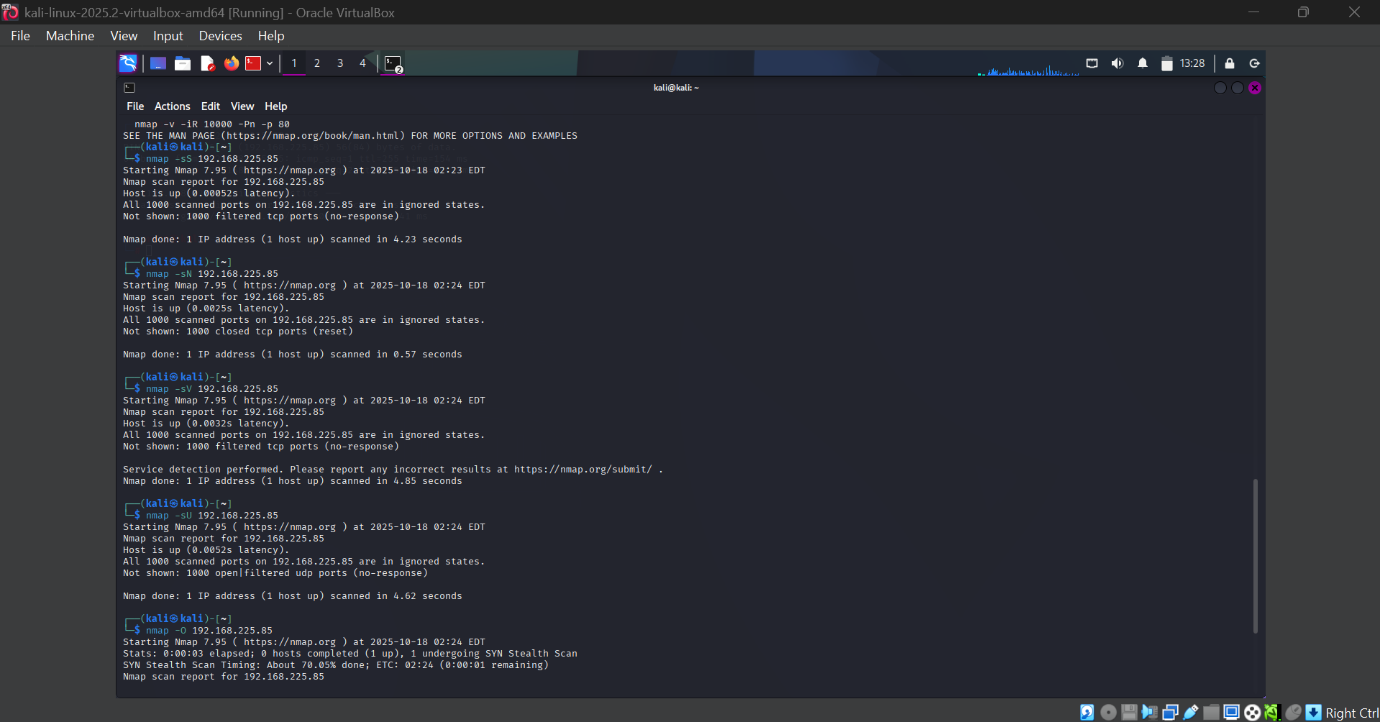
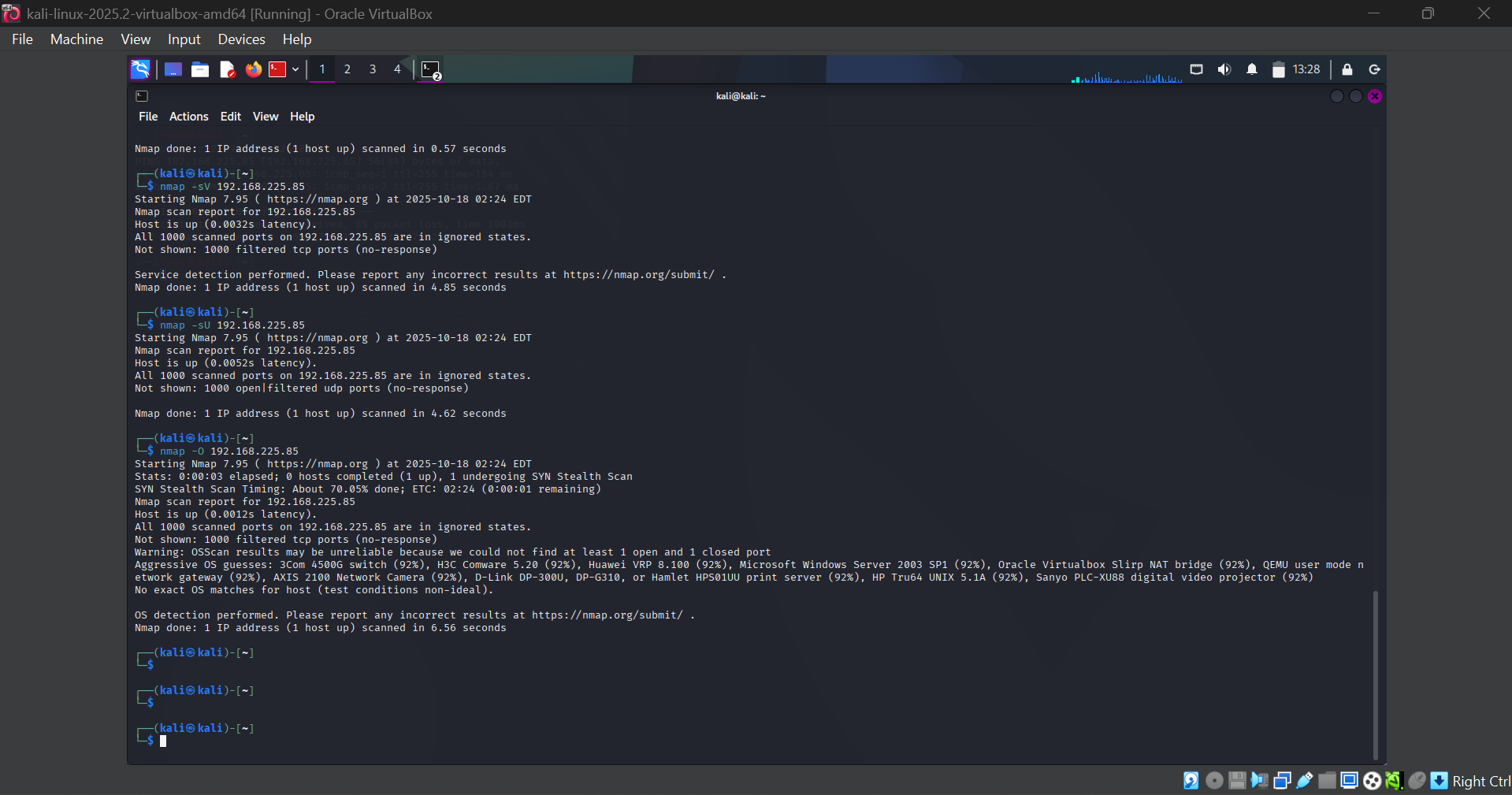
4. nmap -sU 192.168.x.x

* What it does: **UDP** scan — sends UDP probes to discover UDP services (slower & noisier).
* Expected findings: Many ports will show **open filtered** (no reply is ambiguous); when open you might see service names (DNS, SNMP, NTP).
* Quick interpretation: UDP often shows fewer clear responses — if a UDP port is open, it can be a serious vector (e.g., SNMP, DNS).

5. nmap -O 192.168.x.x

* What it does: **OS detection** — analyses packet responses and TTLs to guess the target OS and network device.
* Expected findings: A guessed OS line like OS: Linux 2.6.X with a confidence percentage or “No OS matches” if ambiguous.
* Quick interpretation: Treat as an **educated guess** — useful for triage but confirm with other evidence.

# 3. Visuals of all commands in nmap



# 4. Recommendations

**Recommendations (based on Nmap findings)**

**1. If nmap -sN (NULL scan) shows responsive ports**

* **Action:** Treat any responding ports as potentially open — confirm with a SYN scan (-sS) and service/version scan (-sV).
* **Mitigation:** Close or block unnecessary services at the host or firewall. If the service is required, restrict access by IP and use strong authentication.
* **Hardening:** Disable unused listeners, enable host-based firewall rules (ufw/iptables) and limit exposure to management subnets.

**2. If nmap -sS (SYN scan) reports open ports**

* **Action:** Inventory the services behind open ports and map them to owners/teams. Prioritise critical services (22, 21, 80, 443, 3306, 445).
* **Mitigation:** Immediately close or stop any unapproved services. Move required services behind VPNs, bastion hosts, or NAT so they are not directly exposed.
* **Hardening:** Apply least-privilege access, enforce strong authentication, enable rate-limiting, and deploy network segmentation (zone off lab/demo VMs).

**3. If nmap -sV returns version strings (old/outdated software)**

* **Action:** Match detected versions against vendor advisories/CVEs. Prioritise any with known public exploits.
* **Mitigation:** Patch or upgrade services (prefer supported LTS versions). If patching isn’t possible, remove the service or isolate it from all but required hosts.
* **Hardening:** Disable service banners where possible, remove unnecessary modules, and run the service with minimal privileges. Document versions in an asset inventory.

**4. If nmap -sU shows open/filtered UDP ports (e.g., DNS, SNMP, NTP)**

* **Action:** Validate whether the UDP service is required. UDP services often allow information leakage — treat them as high priority to review.
* **Mitigation:** Disable unused UDP services. For required ones (DNS/SNMP), restrict access to trusted IPs, require authentication where possible (SNMPv3), and limit the query surface.
* **Hardening:** Harden configurations (e.g., rate-limit queries, disable recursion on public DNS, enforce strong SNMP community strings or switch to SNMPv3).

**5. If nmap -O provides an OS fingerprint (or ambiguous result)**

* **Action:** Use the OS guess to confirm patch level and kernel versions on the host. Treat OS detection as guidance — verify on the host.
* **Mitigation:** Ensure the OS is fully patched and supported. Remove unused packages and services that increase attack surface. If the OS is outdated/unpatched, schedule immediate remediation.
* **Hardening:** Apply host hardening guides (CIS benchmarks), enable automatic security updates where safe, and run host-based intrusion detection (OSSEC, Wazuh).

General recommendations

* **Network perimeter & segmentation:** Place lab/demo VMs and vulnerable systems in isolated VLANs with access only from authorized testing hosts. Don’t expose them to production networks.
* **Firewall policy:** Implement deny-by-default firewall rules. Only open ports required for business functions and limit source IP ranges.
* **Authentication & credentials:** Remove default accounts, enforce strong passwords, and prefer key-based auth for SSH. Rotate credentials and avoid embedding secrets in code/config files.
* **Patch management:** Prioritise patching for services flagged by -sV that have known CVEs. Use a staged patch process and maintain a clear patch inventory.
* **Monitoring & alerting:** Log connection attempts and use an IDS/IPS (Snort, Suricata) and SIEM to detect suspicious scanning or repeated authentication attempts.
* **Retest after fixes:** After applying fixes, rerun the same Nmap scans (-sS -sV -sU -O) and an OpenVAS/Nessus scan to validate remediation and ensure no new issues appeared.
* **Documentation & evidence:** For each finding, keep evidence (Nmap output, timestamps, remediation steps taken). That makes the report auditable and reproducible.
* **Ethics & permissions:** Only run active scans on systems you own or have explicit permission to test. Maintain a test authorization record for auditors.