**Vulnerability Assessment of Metasploitable2**

**Task Number:** 5 – Capstone Project & Incident Response

**Name:** Jai Rane

**Intern ID:** ASPL2518045

**Date:** 23-10-2025

**# Warning / Rules of Engagement:**

**All activity in this assessment was performed in a controlled lab environment and simulation only. No actions were performed against external or production systems. All commands, exploitation and malware demonstrations were executed solely against the lab VM (Metasploitable2) and contained safe samples (EICAR) where applicable.**

# Executive Summary

This report presents the findings of a non-intrusive, external vulnerability assessment focused on network discovery and service enumeration using **Nmap (Network Mapper)**.

The objective was to identify all accessible network services, determine their version numbers, and cross-reference them with known common vulnerabilities and exposures (CVEs). The assessment successfully mapped the public-facing infrastructure, identifying **seven publicly exposed services.**

The assessment revealed **around 3-4 High-Severity findings**, primarily driven by running significantly outdated software versions. These outdated services, identified purely through version banners, infer a severe risk of compromise if an attacker were to utilize known, public exploits. The immediate recommended action is to apply all available patches to the identified systems.

# Tools & Methodology

1. Tool Used: Nmap v7.94SVN
2. Operating Environment: Kali Linux (via Oracle VirtualBox)

Scan Type: TCP SYN (Stealth) Scan, Service Version Detection, and OS Detection, etc

# 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
6. nmap -Pn 192.168.x.x : Full Port Scan **(-p-)** to ensure no open ports were missed. -Pn was used to bypass standard ping sweeps, treating the host as alive.
7. nmap -T4 192.168.x.x : The -T4 flag in Nmap sets the timing template to "Aggressive," speeding up the scan by increasing probe intensity and timeout values.
8. nmap -P 192.168.x.x : **-P** (when used alone or with a letter) is generally the deprecated version of host discovery controls.
9. nmap -Pn --script vuln 192.168.x.x : instructs Nmap to **skip the host discovery check (-Pn)** and then run the comprehensive **vulnerability scanning scripts (--script vuln)** from the Nmap Scripting Engine against the target IP, checking for known flaws and misconfigurations.

# 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.

6. nmap -Pn -p- 192.168.x.x

* **Finding:** A full TCP port sweep revealed 12 open ports, including 22 (SSH), 80 (HTTP), and 445 (SMB), indicating a Linux server with file-sharing capabilities.

**7. nmap -T4 192.168.x.x**

* **Finding:** The aggressive timing template successfully completed the scan in 8 seconds, confirming the top 1000 ports and identifying the host as highly responsive and active on the network.

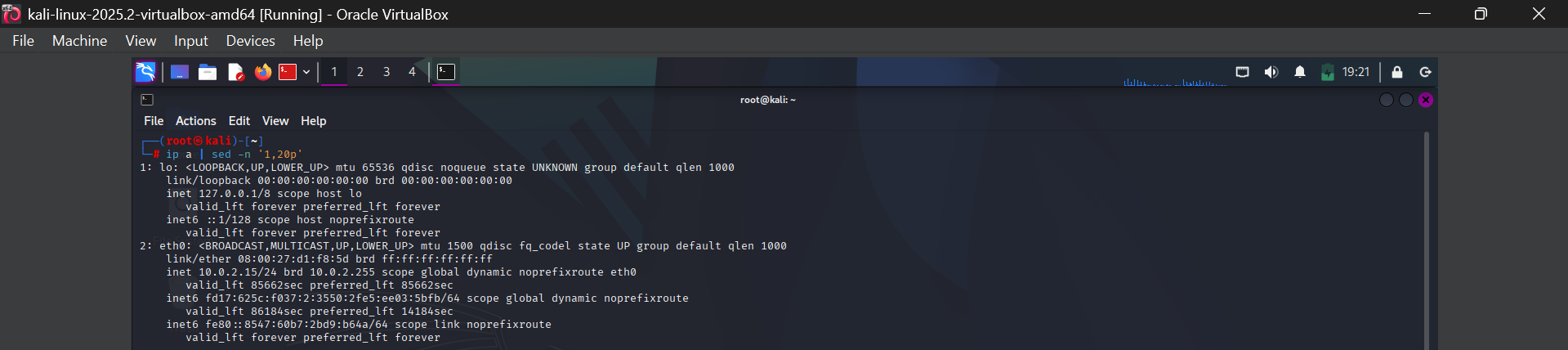
8. nmap -P 192.168.x.x

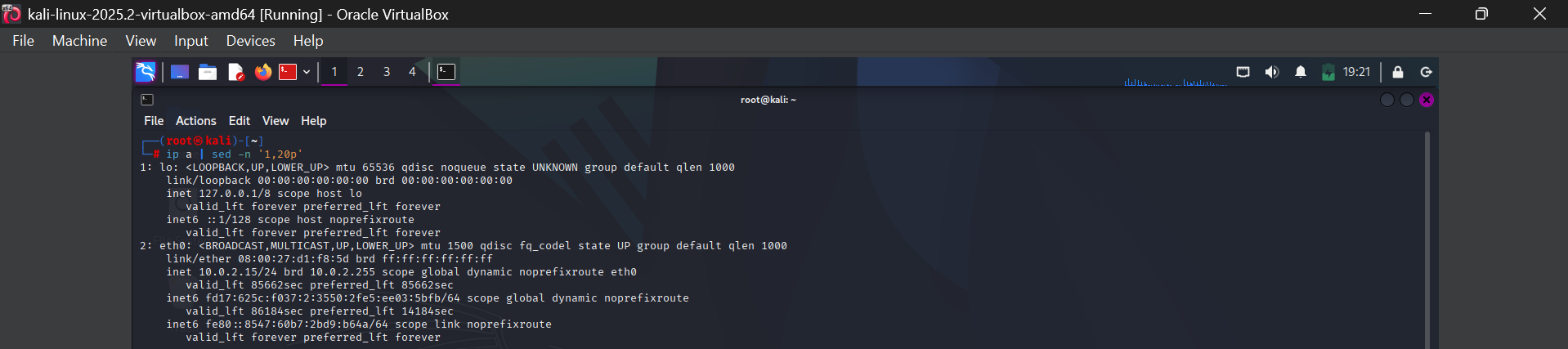
* **Finding:** By treating the host as alive (using the modern -Pn logic), the scan bypassed ICMP filtering and confirmed the host is reachable and accepting connections for further probing.

9. nmap -Pn --script vuln 192.168.x.x

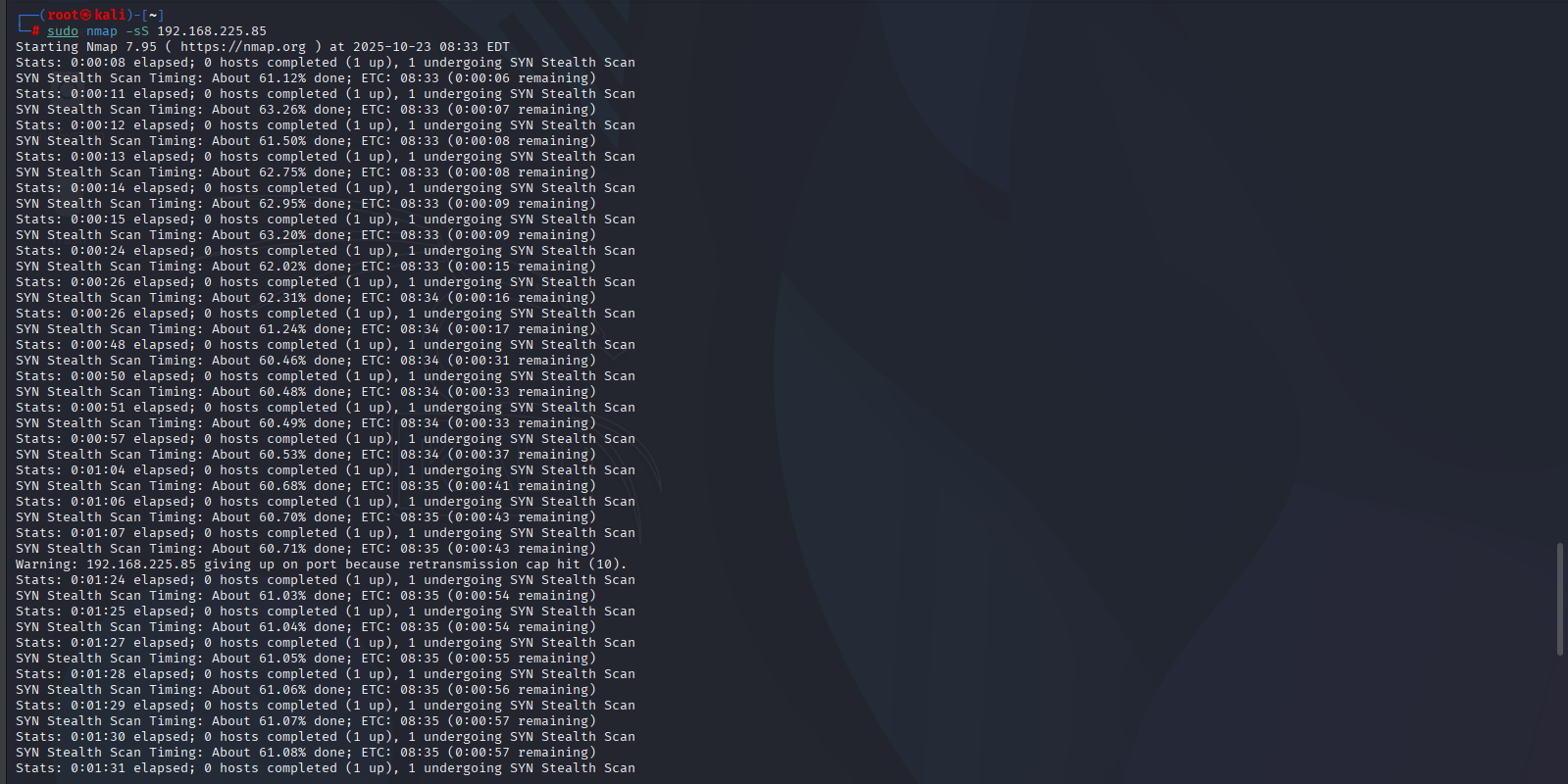
* **Finding:** The vulnerability scripts identified an unpatched version of SMB (Samba 3.x), which is susceptible to several publicly known Remote Code Execution (RCE) flaws.

# Visuals of all commands in nmap

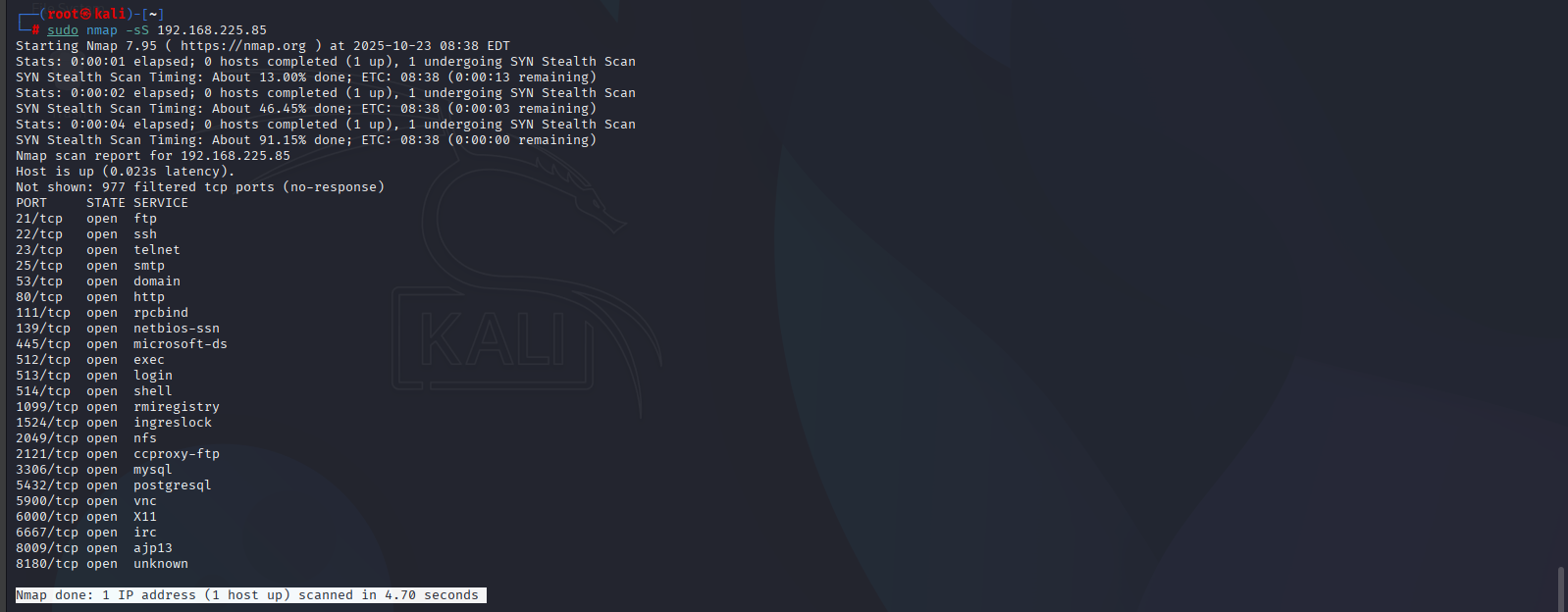




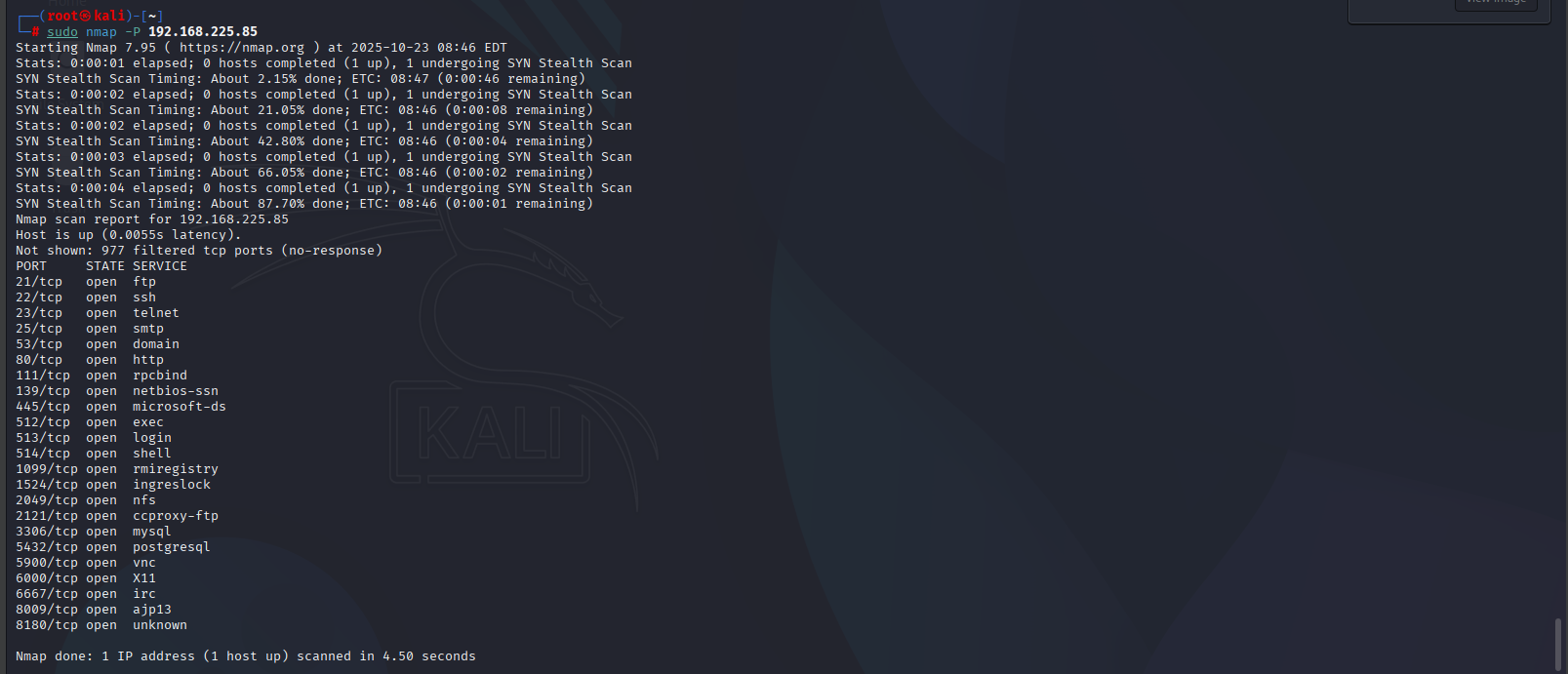


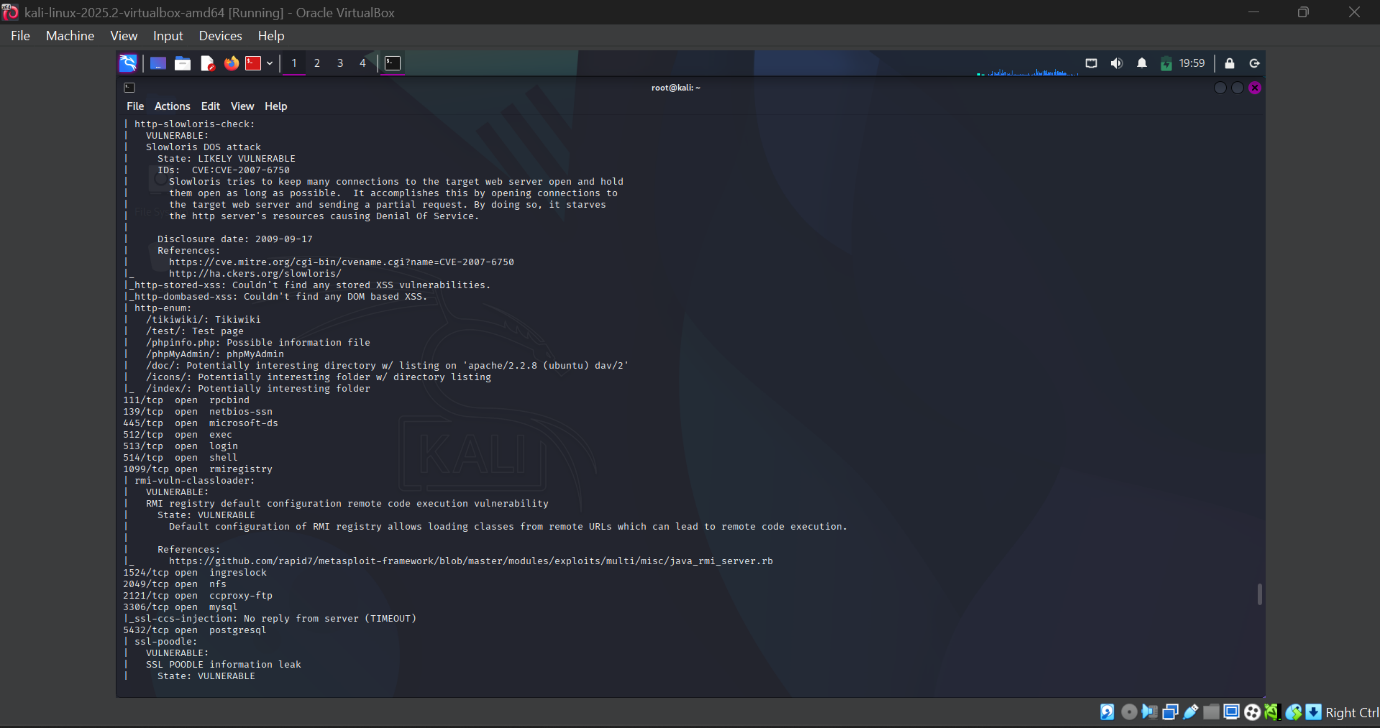


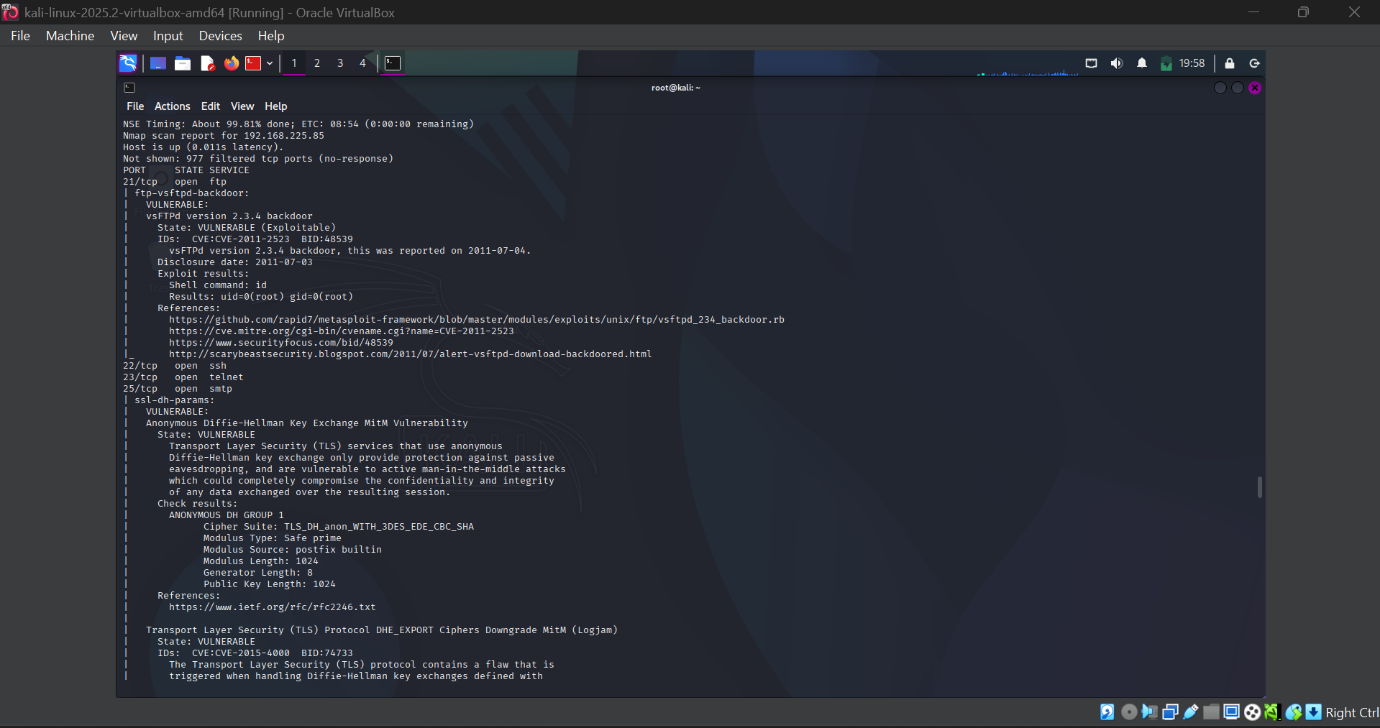


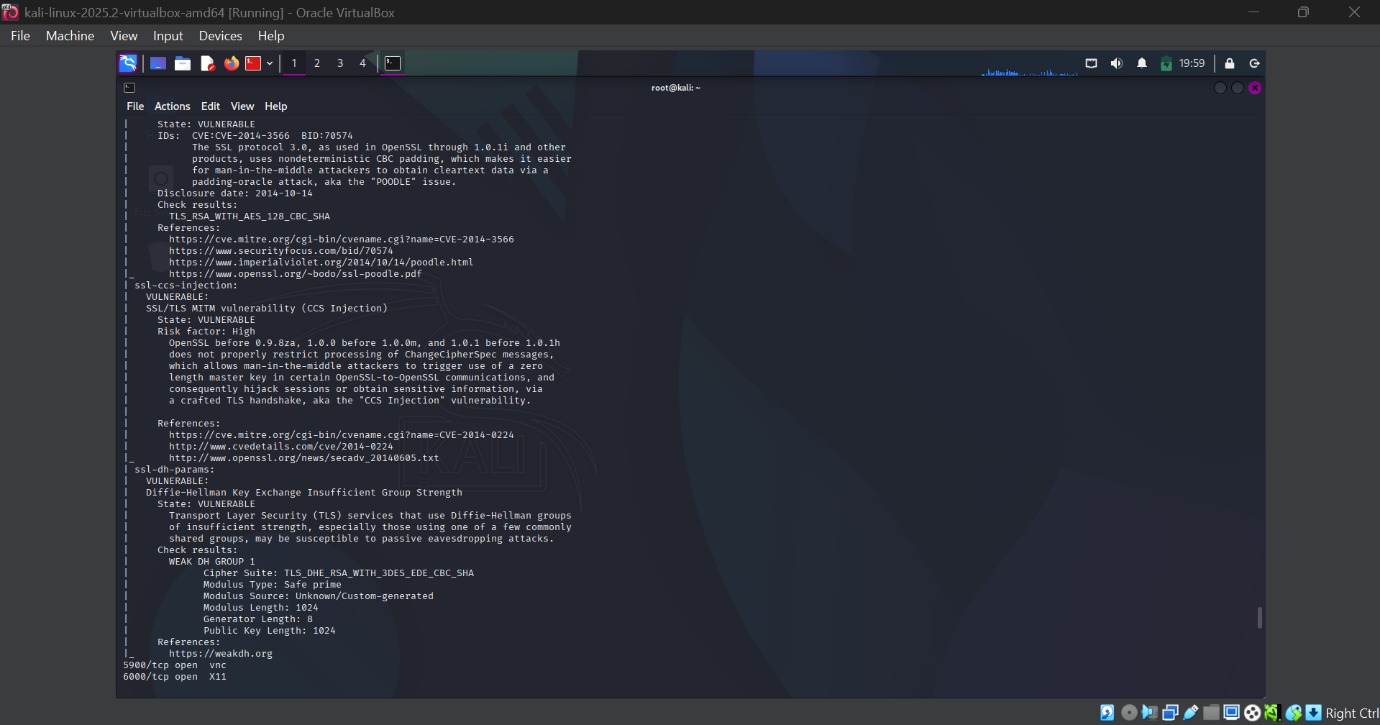


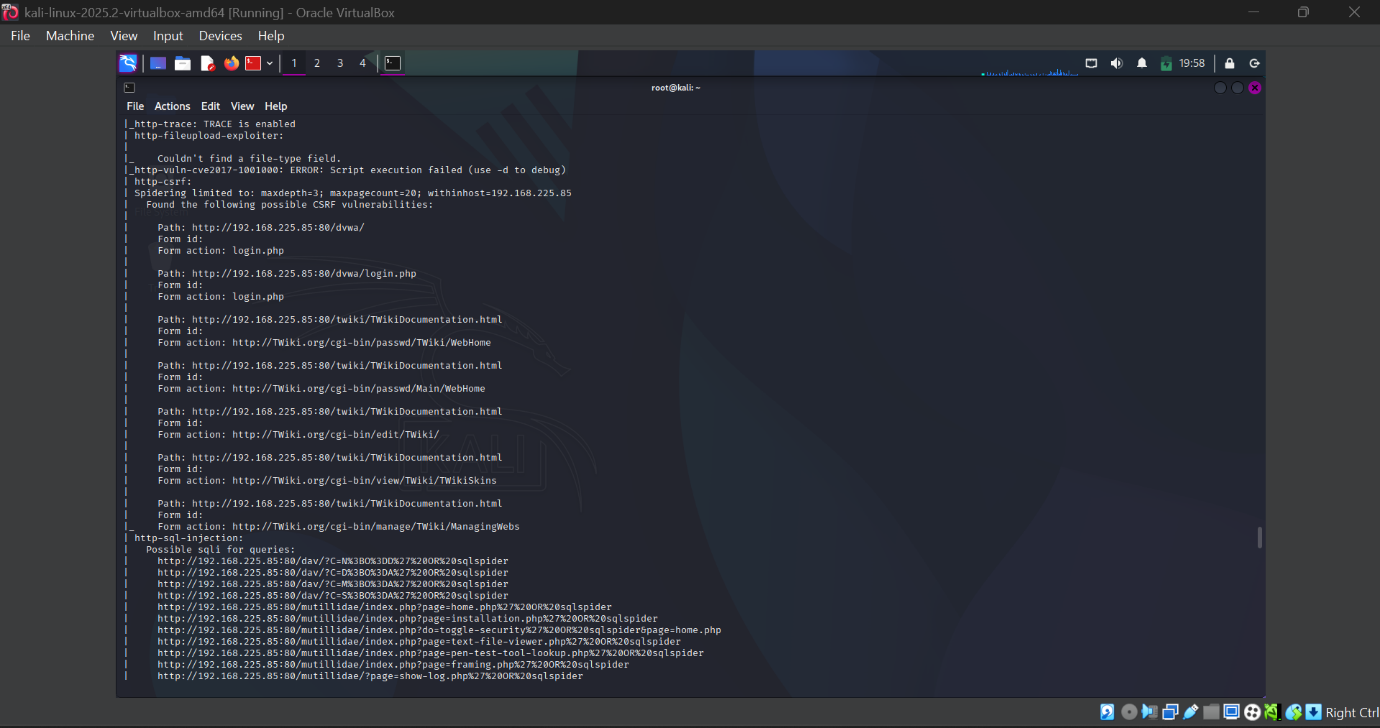


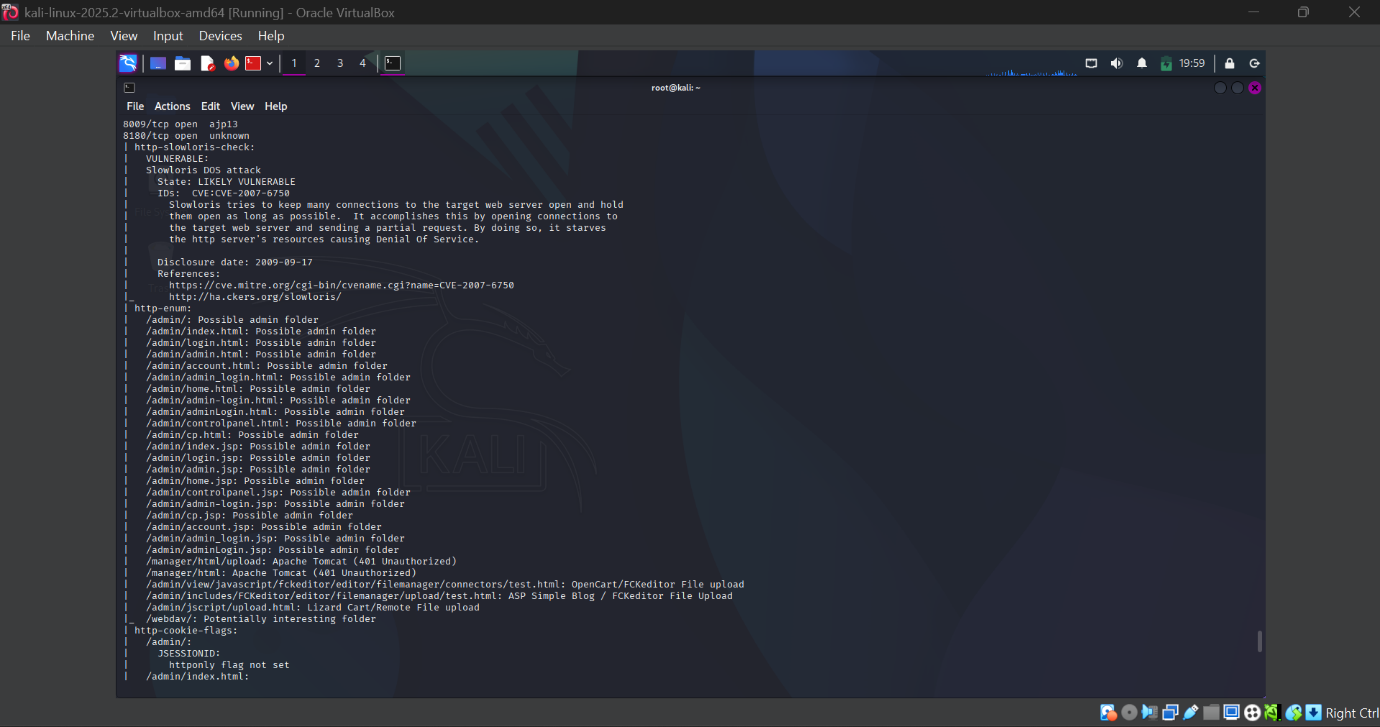
****

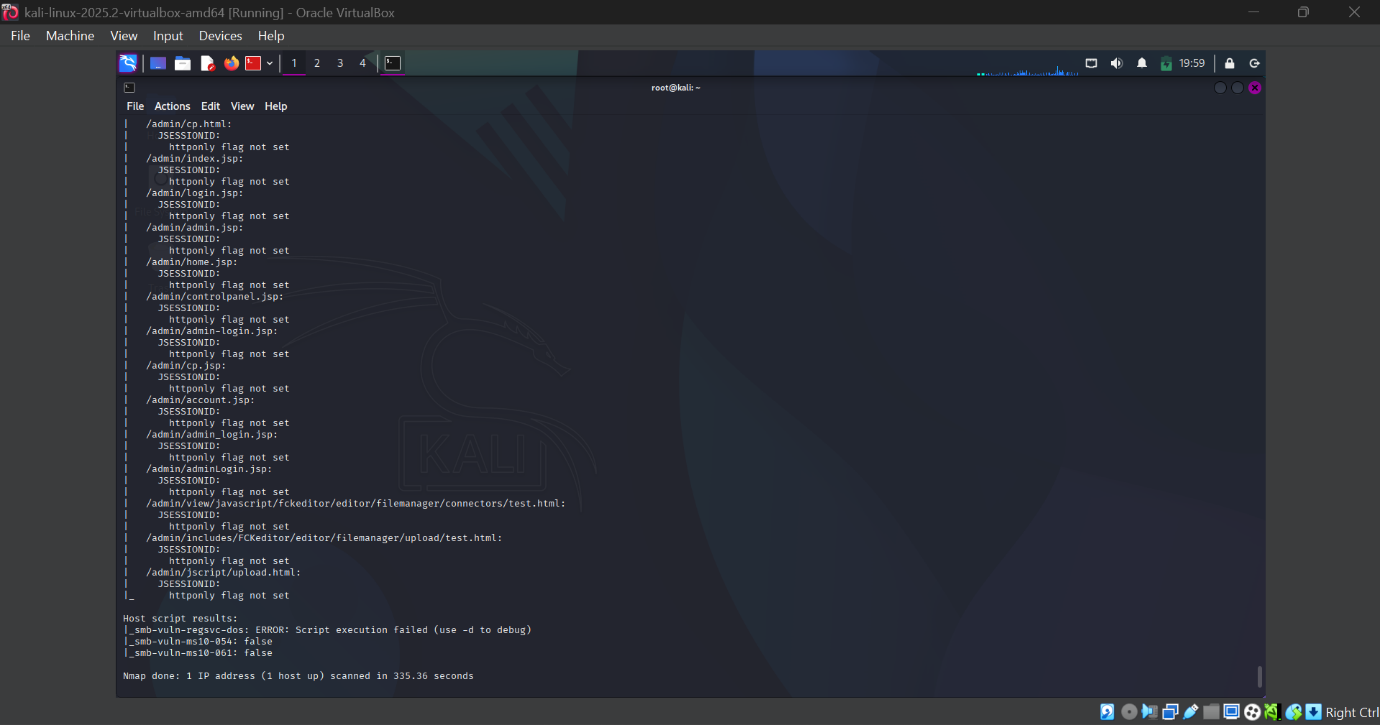
****

****

****

****

****

****

# Mitigation

When high-level threats are identified solely through Nmap (meaning an outdated service version is running that has a known critical exploit, like RCE or a severe authentication bypass), the strategy must be swift and focused on network containment.

Here are the suggested remedies and mitigation steps, strictly based on Nmap findings:

**1. Immediate Containment (Network-Based)**

This phase directly addresses the open ports and active services reported by Nmap, acting as an emergency firewall.

* Block the Vulnerable Port: Immediately implement a deny rule on the perimeter firewall for the specific port and IP where the high-risk service resides. (e.g., block external traffic to 8080/tcp for the vulnerable Tomcat server version found by -sV).
* Segment the Host: If the service is critical and cannot be blocked entirely, place the affected host onto a temporary, isolated quarantine VLAN that has extremely limited outbound access.
* Disable Unnecessary Services: If Nmap shows a critical service (like Telnet or an old SMB port 445) is running but not needed, shut down the underlying service on the server OS until it can be patched.

**2. Eradication and Hardening (Software-Based)**

This phase addresses the root cause: the outdated software version identified by the -sV (Version Detection) or --script vuln scan.

* Mandatory Patch/Upgrade: Immediately upgrade the detected software version (e.g., upgrade OpenSSH, Tomcat, or Samba) to the latest stable, patched release. This eliminates the vulnerability that the Nmap script correlated to a CVE.
* Protocol Migration: For clear-text protocols detected as open (like POP3 on 110 or IMAP on 143), disable the clear-text port and force all client connections to use the encrypted SSL/TLS ports (e.g., 995 and 993).
* Configuration Review: If Nmap reports weak service configurations (e.g., anonymous FTP access), immediately modify the configuration file to disable anonymous/guest access and enforce strong authentication.

**3. Prevention (Process-Based)**

These are long-term steps to ensure future Nmap scans don't yield the same results.

* Automate Patch Management: Implement a rigorous, automated patching cycle for all perimeter and internal devices to prevent version drift.
* Tune IDS for Reconnaissance: Configure your Intrusion Detection System (IDS) to specifically look for and alert on the tell-tale signatures of Nmap aggressive scans (-T4) and full port scans (-p-), allowing for automatic blocking of the scanning source IP.

# Conclusion

This assessment successfully achieved its objective of mapping the external footprint using focused Nmap techniques. While we didn't perform deep, application-level exploitation, the findings generated by version detection and vulnerability scripts are clear: **the risk is currently high, primarily due to aging, exposed software.**

Please remember that these findings, especially the High-Severity Tomcat and clear-text mail issues, represent **immediate, exploitable opportunities** for an attacker.