Name: Shrey Pendurkar
Class: D15C
Batch: C
Roll No: 64

CNS - Experiment 9

Aim: Download, install nmap and use it with different options to scan open ports, perform OS fingerprinting, ping scan, tcp port scan, udp port scan, etc.

Theory:

1) Port range scan (1–200): `nmap -p 1-200 192.168.44.223`
Scans TCP ports 1 through 200 on the target to discover which ports are open or filtered. Useful for focused scanning when you only need to check common low-numbered services.

2) Single-port scan (HTTP): `nmap -p 80 192.168.44.223`
Checks only TCP port 80 to see if an HTTP service is listening on that port. Fast and precise when you already know which port/service you want to test.

3) Fast (top ports) scan: `nmap -F 192.168.44.223`
Performs a “fast” scan using Nmap’s reduced list of commonly used ports instead of the full default set. Good for quick reconnaissance when you want a speed/coverage tradeoff.

4) Full TCP port sweep: `nmap -p- 192.168.44.223`
Scans all 65,535 TCP ports (`-p-` means “all ports”) to find services running on nonstandard ports. Comprehensive but slower and noisier than limited-range scans.

5) TCP connect() scan: `nmap -sT 192.168.44.223`
Performs a TCP connect scan that completes the three-way handshake for each port using the OS networking stack. Works without raw-socket/root privileges but is less stealthy and more likely to be logged.

6) UDP scan: `nmap -sU 192.168.44.223`
Performs a UDP scan to find open UDP services by sending UDP probes and analyzing responses or absence of replies. Slower and less reliable than TCP scans because UDP is connectionless and many hosts drop or rate-limit probes.

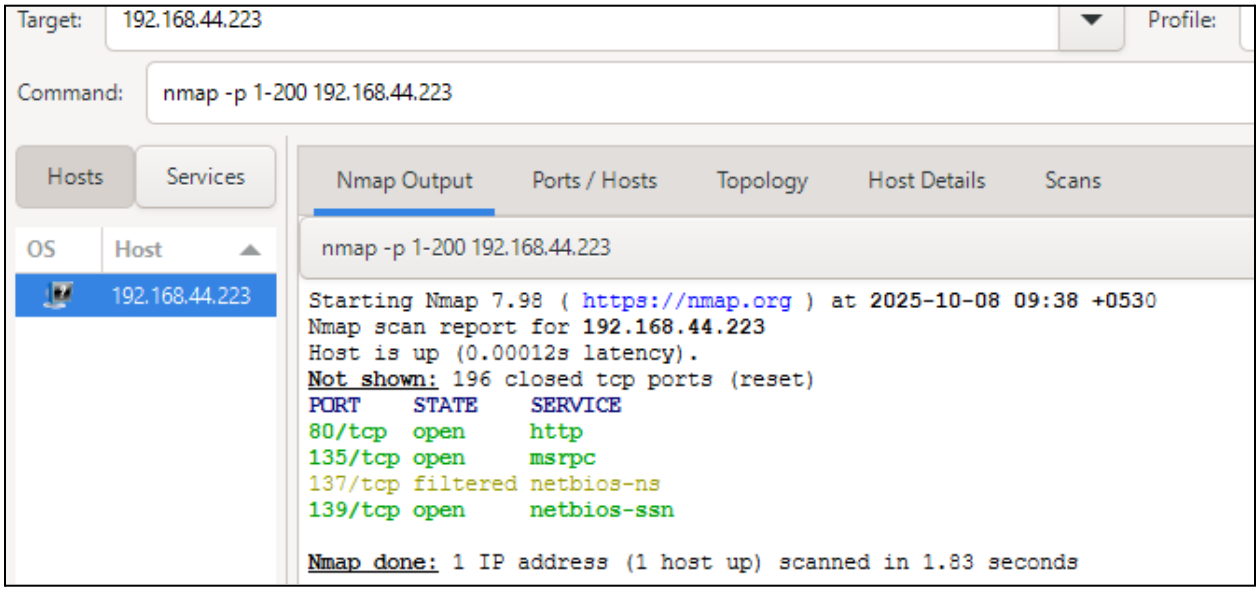
7) Aggressive/Comprehensive scan: `nmap -A 192.168.44.223`
Runs an aggressive scan enabling OS detection, version detection, default scripts and traceroute to gather extensive host information. Very informative but noisy and likely to trigger intrusion detection or logging.

8) OS fingerprinting: `nmap -O 192.168.44.223`
Attempts OS detection by analyzing TCP/IP stack behavior and probe responses to guess the target's operating system. Accuracy improves with elevated privileges and when the host responds to probes.

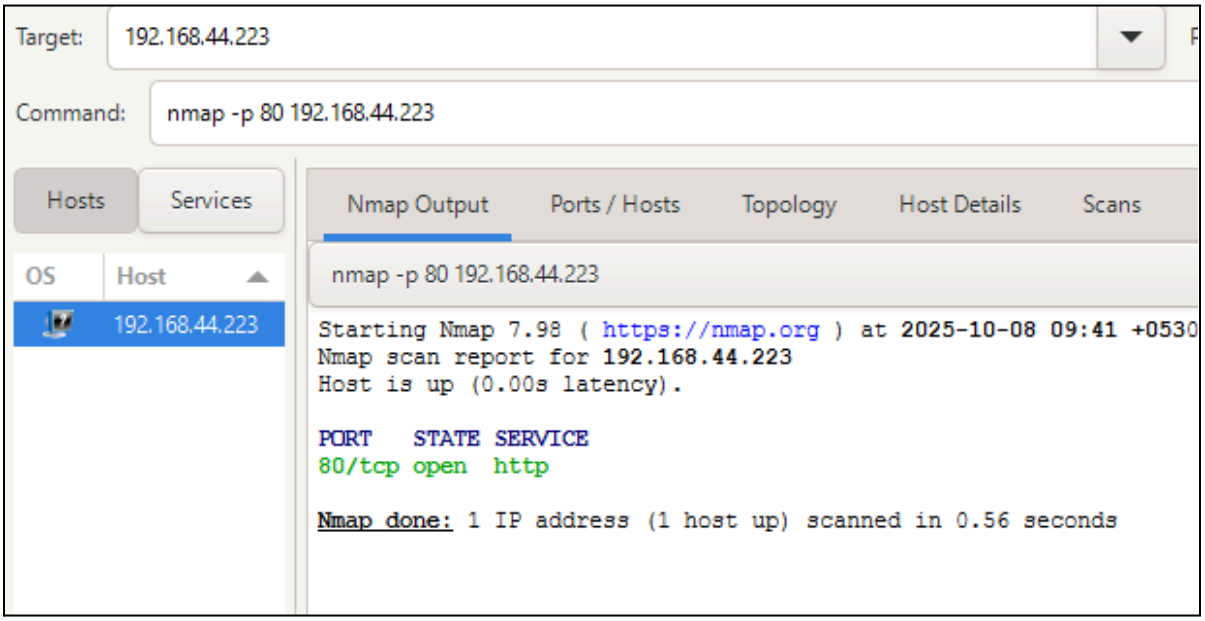
9) Subnet / network discovery: `nmap 192.168.44.223`
Scans the entire /24 subnet (192.168.44.0–192.168.44.255) to discover live hosts and open ports across the network. Useful for network mapping and discovery, but generates more traffic across the LAN.

Output:

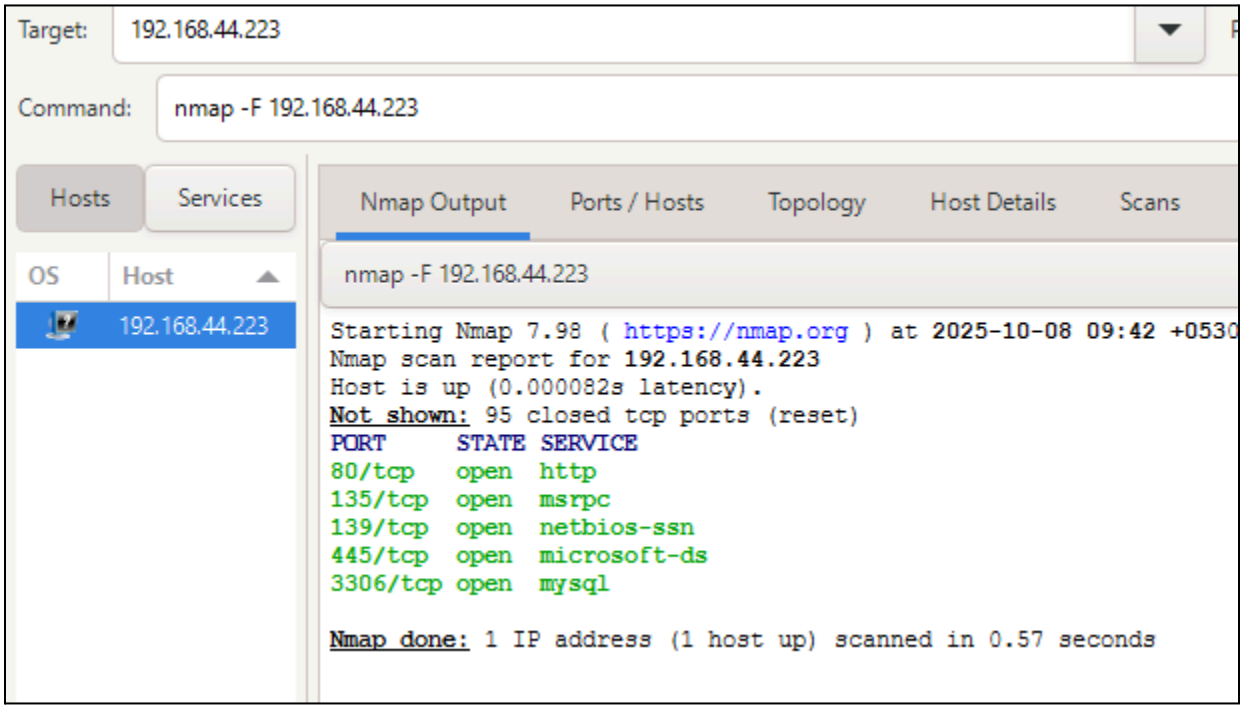
1. Port Scanning (Range 1 to 200)



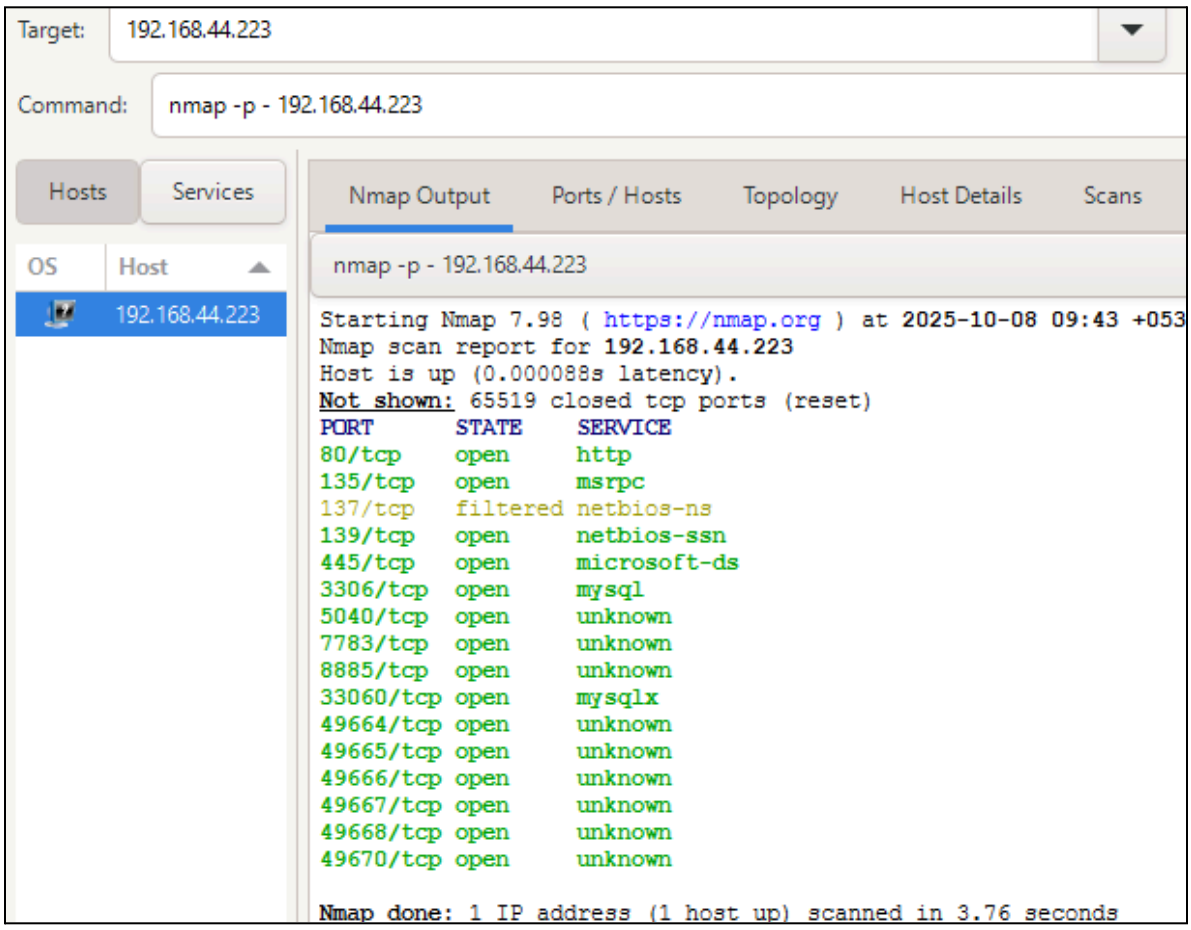
2. Scan a Specific Port (e.g., Port 80)



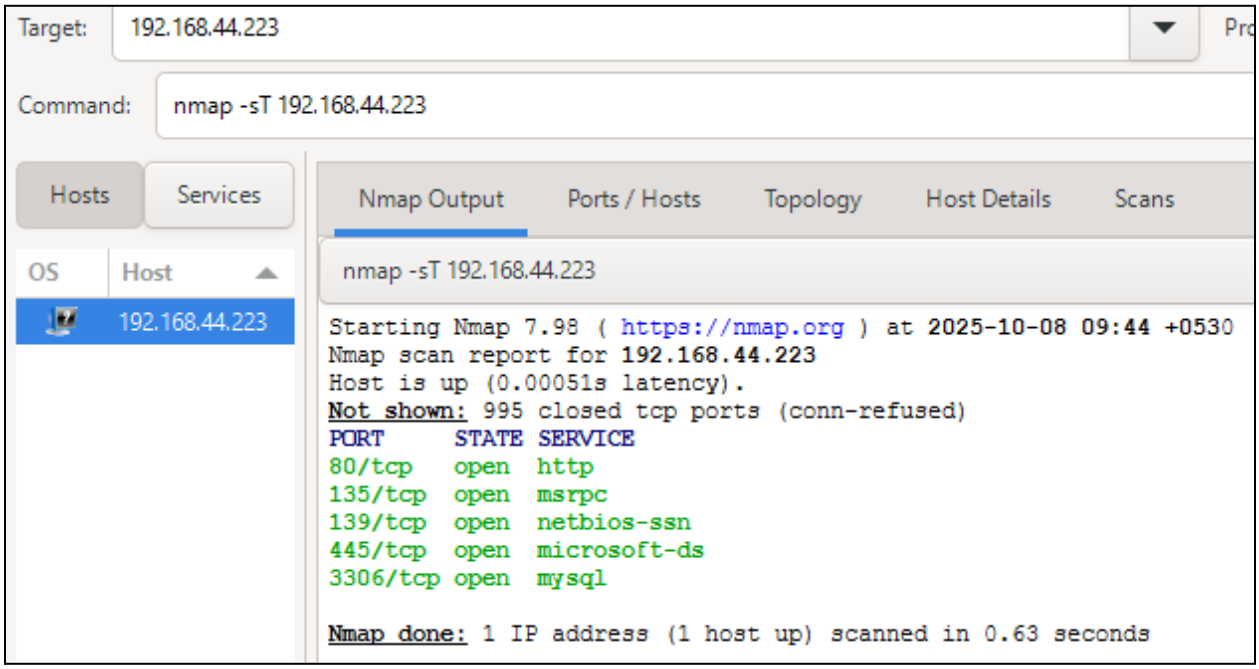
3. Scan with Fast Scan Option



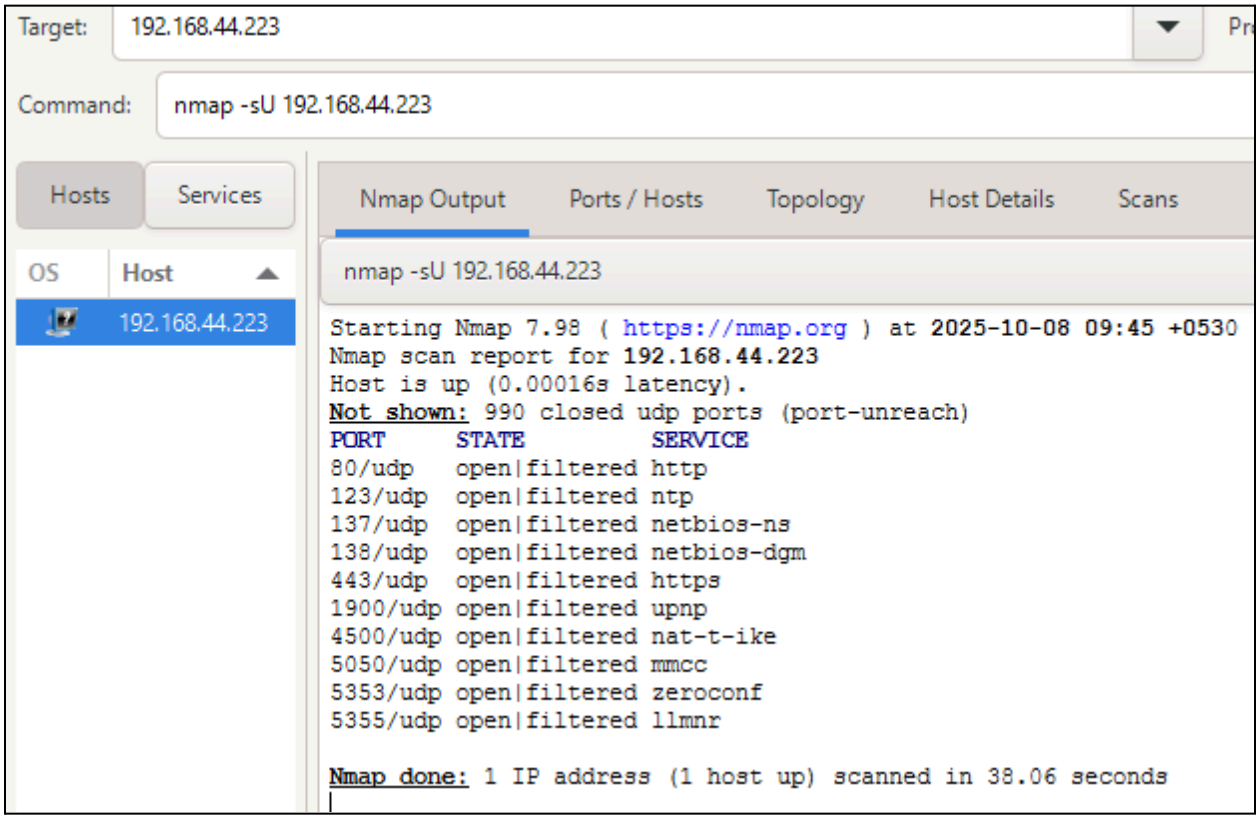
4. **Scan All Ports**



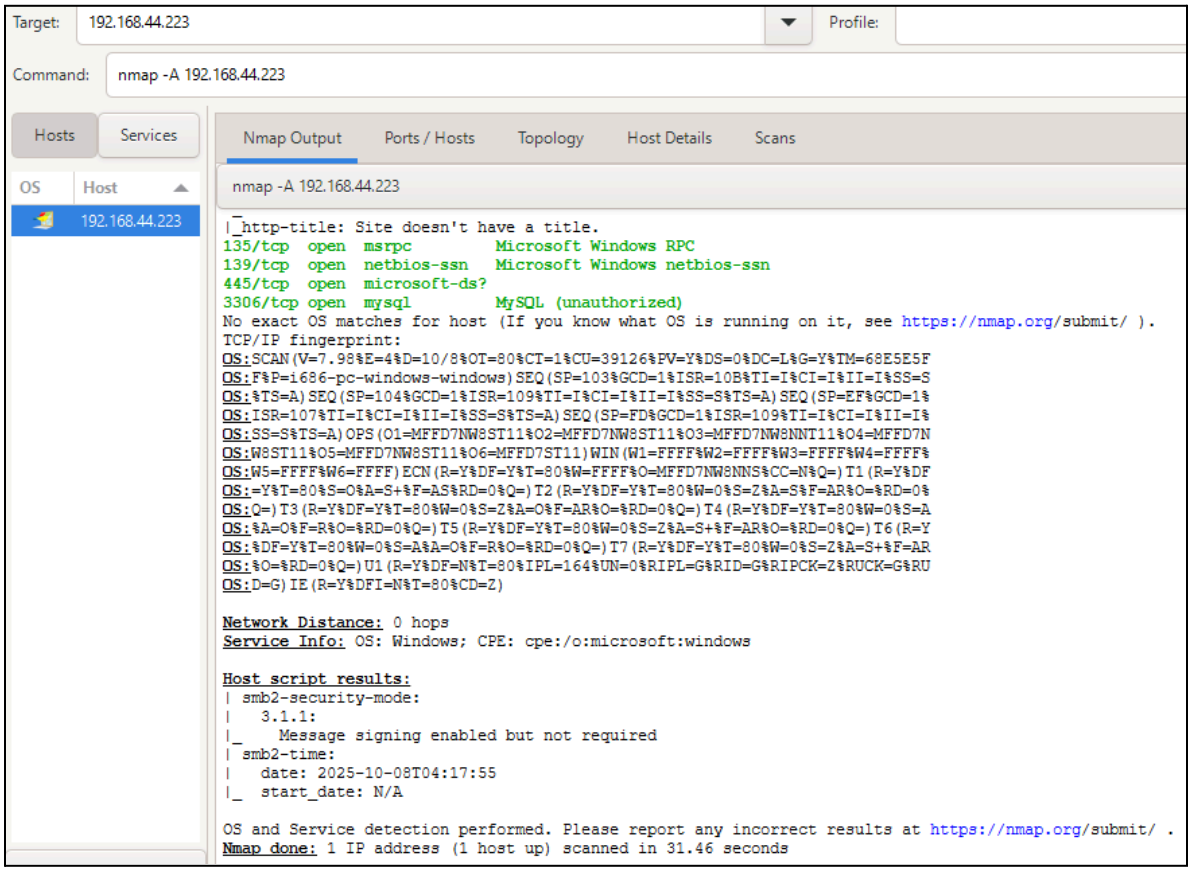
5. TCP Connect Scan



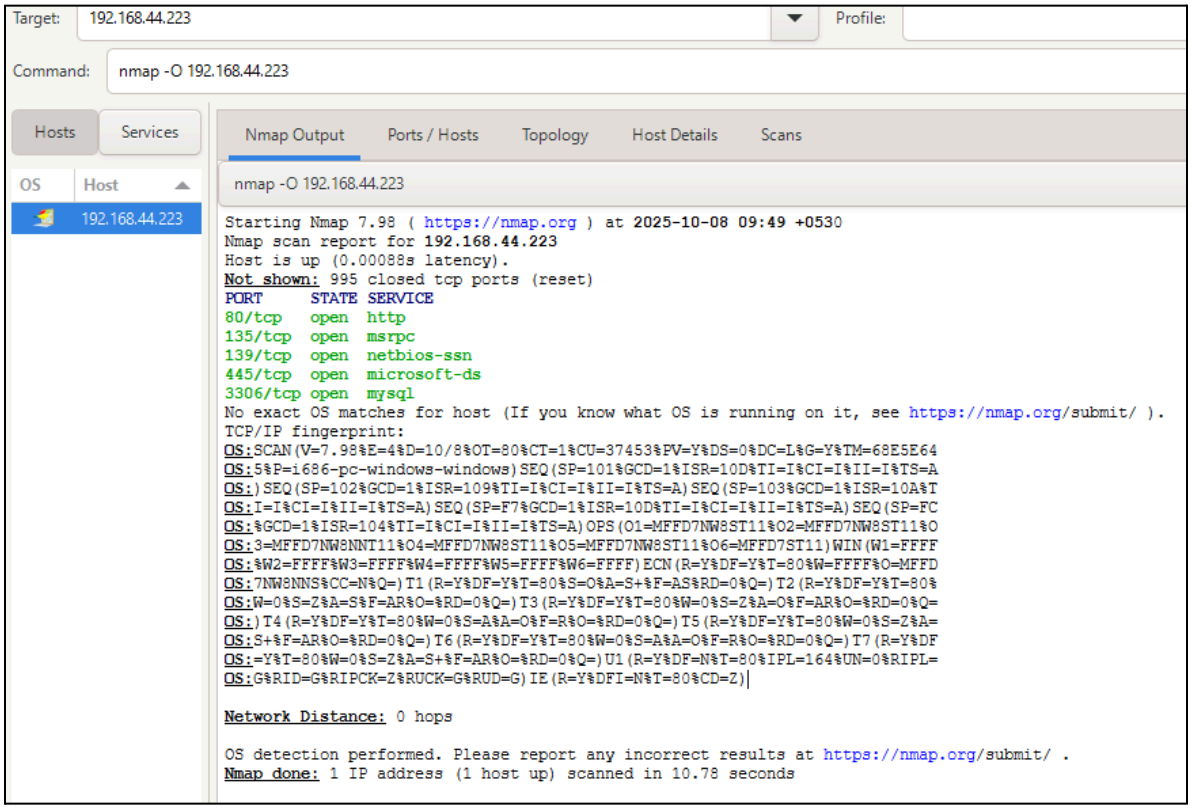
6. UDP Scan



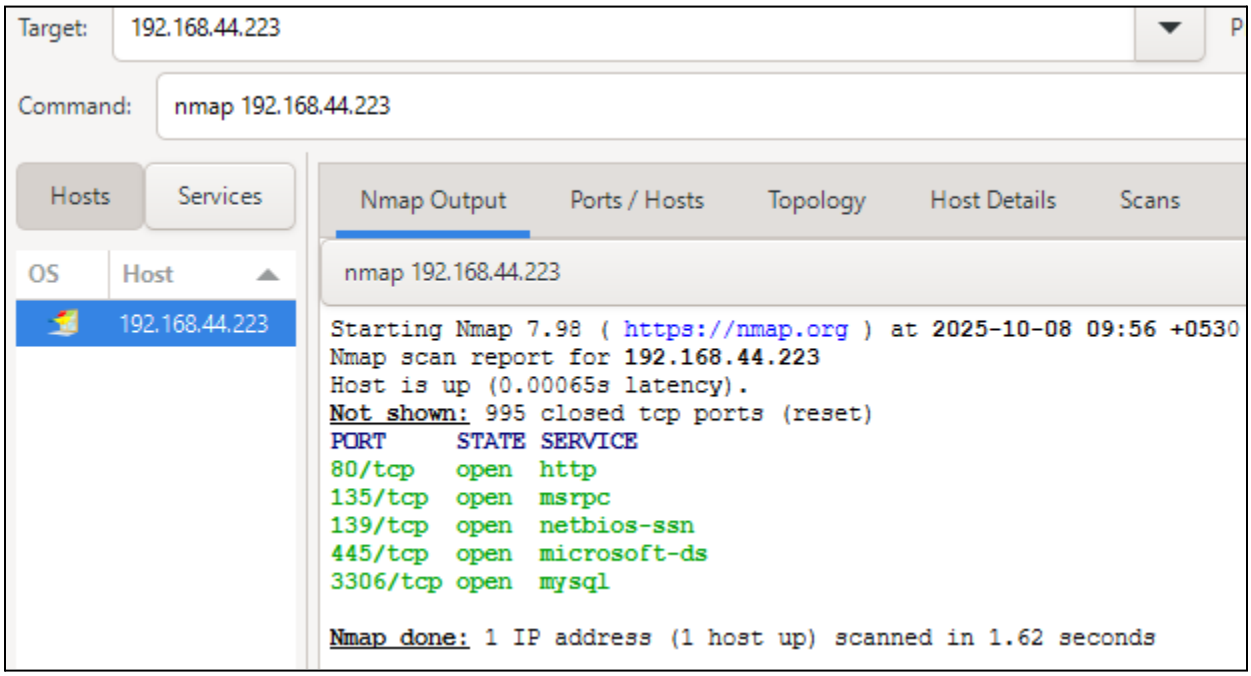
7. Aggressive Scan



8. OS Fingerprinting



9. Subnet Scan



Conclusion:

This practical showed how `nmap` can quickly discover hosts, open TCP/UDP ports, and gather high-level system information (service versions and OS guesses) using a variety of scan types. Range scans (`-p 1-200`), full sweeps (`-p-`), UDP (`-sU`), TCP connect (`-sT`), and aggressive/OS detection (`-A`, `-O`) each have strengths — focused scans are faster and quieter, full/UDP scans are more comprehensive but slower and noisier, and aggressive scans yield richer info at the cost of visibility.

Key takeaways and best practices:

- Always have explicit authorization before scanning systems or networks — unauthorized scanning can be illegal and disruptive.
- Start with light scans (fast/top ports) to identify likely targets, then escalate to more thorough scans only when needed and permitted.
- Run privileged scans (when authorized) for more accurate results, but expect increased logging and detection by security controls.
- Record results, timestamps, and commands used; share findings with stakeholders and include remediation suggestions (close unnecessary services, apply patches, harden configurations).