**Experiment 8**

**Aim**: Demonstrate the use of NMAP with different options to scan open ports, perform OS fingerprinting ,ping scan, tcp port scan ,udp port scan, etc

**Learning Objectives for Using Nmap:**

1. **Understand the Basics of Network Scanning**: Gain foundational knowledge of network scanning and how tools like Nmap are used to discover hosts, services, and potential vulnerabilities.
2. **Master Nmap Syntax and Options**: Learn how to use various Nmap options to tailor scans to specific needs, including scanning for open ports, performing OS fingerprinting, and distinguishing between TCP and UDP protocols

**Theory:**

Nmap (Network Mapper) is a versatile open-source tool used for network discovery and security auditing. Understanding the theory behind its various scanning techniques is crucial for effectively utilizing the tool and interpreting its results. Here’s an overview of the key scanning methods and their theoretical foundations:

1. Port Scanning

**Purpose**: Identify open ports on a target host to determine which services are running and potentially vulnerable.

**TCP Connect Scan (-sT)**:

* + **Mechanism**: Establishes a full TCP connection with the target ports. It sends a SYN

(synchronize) packet, and if the port is open, the target responds with a SYN-ACK (synchronize-acknowledge) packet. The scanner then sends an ACK (acknowledge) to complete the handshake.

* + **Theory**: This scan method is straightforward but can be easily detected and logged by the target system, as it completes the TCP handshake.

**SYN Scan (-sS)**:

* + **Mechanism**: Sends SYN packets to target ports. If the port is open, the target responds with SYN-ACK. The scanner then sends an RST (reset) packet to abort the handshake before it completes.
  + **Theory**: Also known as a "half-open" scan, it is stealthier than a TCP connect scan because it does not establish a full connection, making it less likely to be logged by the target.

1. OS Fingerprinting

**Purpose**: Determine the operating system running on a target host.

**Mechanism**:

* + **Theory**: OS fingerprinting relies on analyzing network responses to various probes. Different operating systems respond uniquely to specific packets or sequences of packets due to variations in TCP/IP stack implementations. Nmap uses a database of known responses and patterns to make educated guesses about the target OS.

1. Ping Scan (-sn)

**Purpose**: Determine which hosts are up and responding on a network without scanning ports.

**Mechanism**:

* + **Theory**: This scan sends ICMP Echo Request (ping) packets, or sometimes uses other methods like TCP SYN packets to ports like 80 or 443, to see if the target responds. If a response is received, the host is considered "up."
  + **Note**: Some hosts may be configured to ignore ICMP requests, so additional methods may be used to determine host availability.

1. UDP Scanning (-sU)

**Purpose**: Identify open UDP ports on a target system.

**Mechanism**:

* + **Theory**: UDP scanning sends UDP packets to the target ports and waits for responses. If a port is closed, the target typically responds with an ICMP Port Unreachable message. If there’s no response, the port may be open or filtered (firewalls or other security devices might drop the packet without response).
  + **Challenges**: UDP scanning can be less reliable and slower than TCP scanning due to the connectionless nature of UDP and potential filtering by firewalls.

1. Combining Scans

**Purpose**: Perform a comprehensive scan that combines multiple scanning techniques for a detailed network assessment.

**Mechanism**:

* + **Theory**: Combining different scan types, such as TCP SYN and UDP scans, provides a fuller picture of the target's network services and vulnerabilities. This approach allows for simultaneous assessment of both TCP and UDP services and can help in identifying services that might be missed by scanning one protocol type alone.

**OS fingerprinting**

**OS Fingerprinting** is a technique used to determine the operating system (OS) running on a

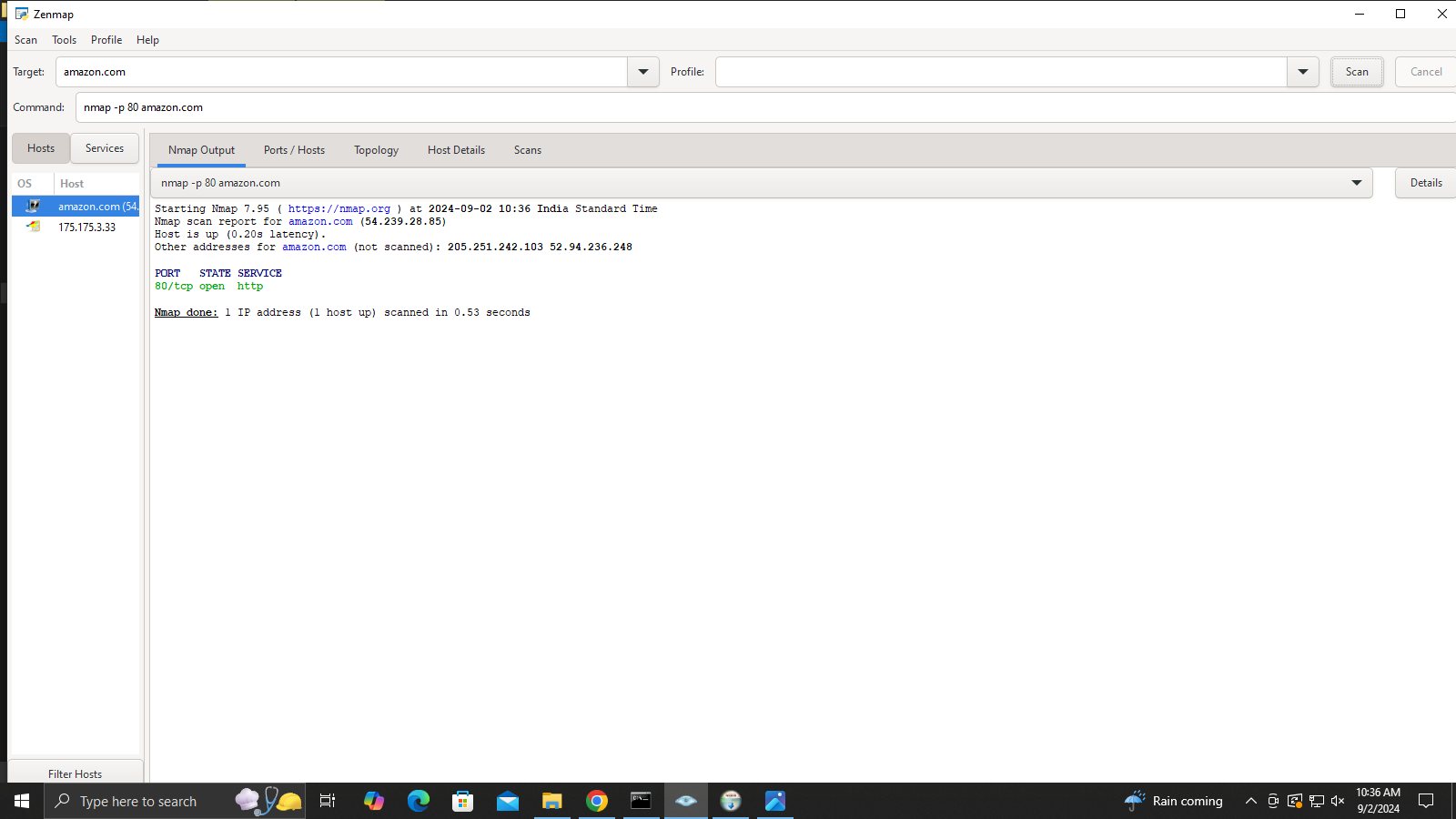
networked device. Nmap’s OS fingerprinting capabilities help identify the OS by analyzing how a target responds to various network probes and TCP/IP stack behaviors.

How OS Fingerprinting Works

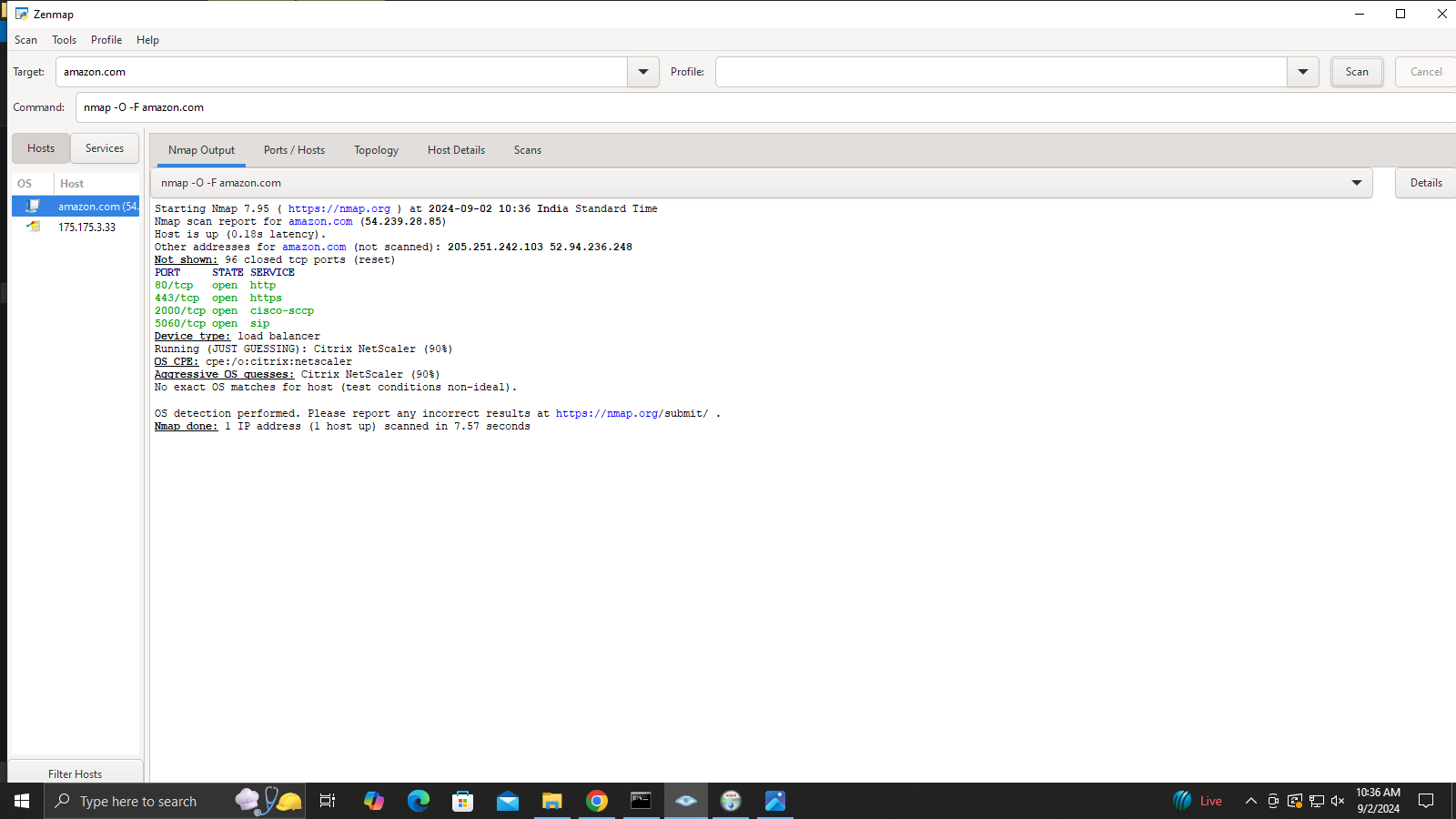
* 1. **Probe Packets**: Nmap sends a series of specially crafted packets to the target system. These packets vary in their headers and content to probe different aspects of the network stack.
  2. **Response Analysis**: The target system responds to these probes in ways that are influenced by its OS and network stack implementation. Responses may include variations in TCP options, packet timing, and the handling of unusual or malformed packets.
  3. **Pattern Matching**: Nmap compares the responses it receives to a database of known OS signatures. This database contains patterns and behaviors characteristic of different operating systems and versions.
  4. **Result Reporting**: Based on the comparison, Nmap estimates the most likely OS and version running on the target system. It may also provide additional details about the system

**Implementation:**

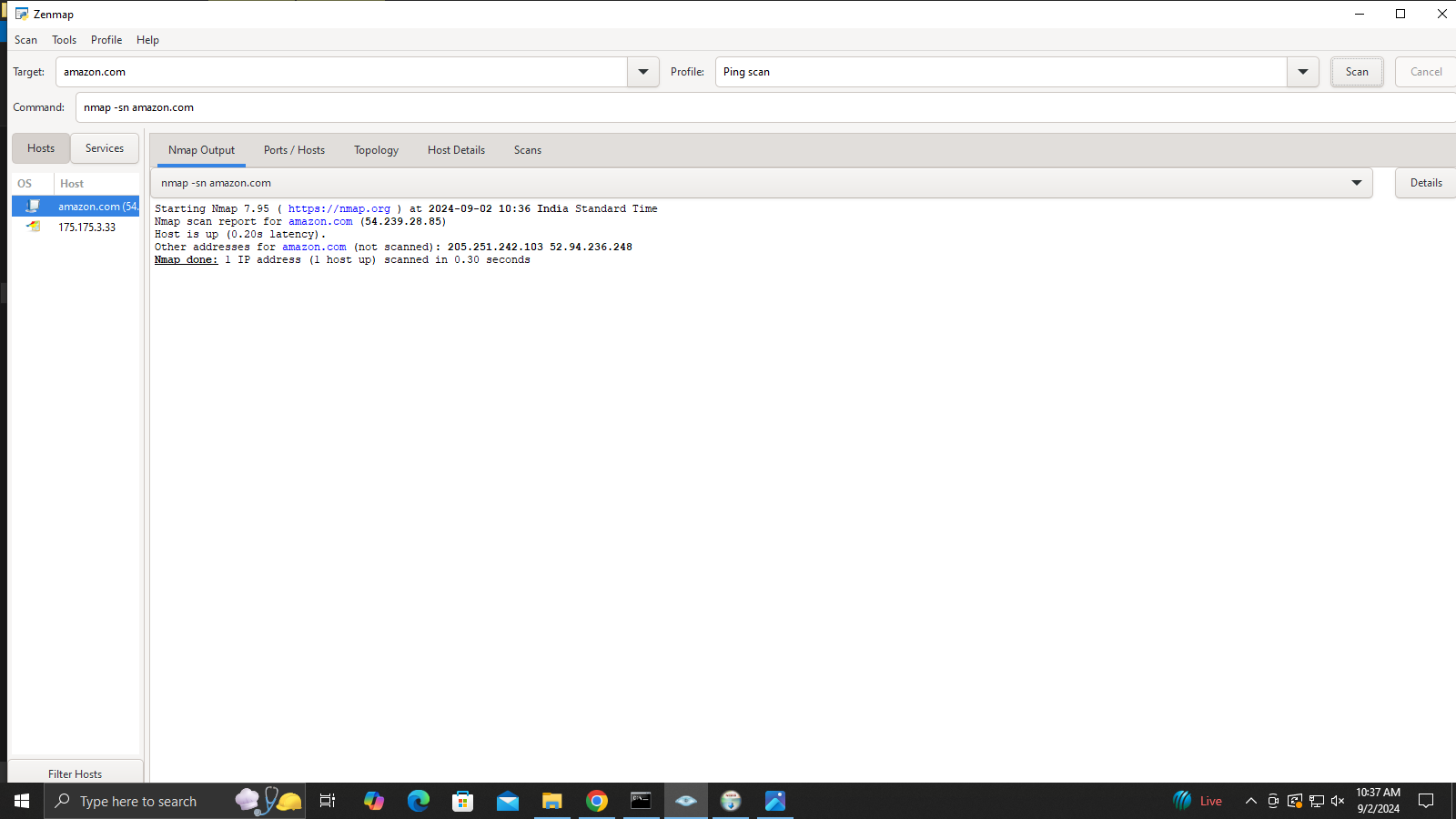
1. **To scan open ports**

****

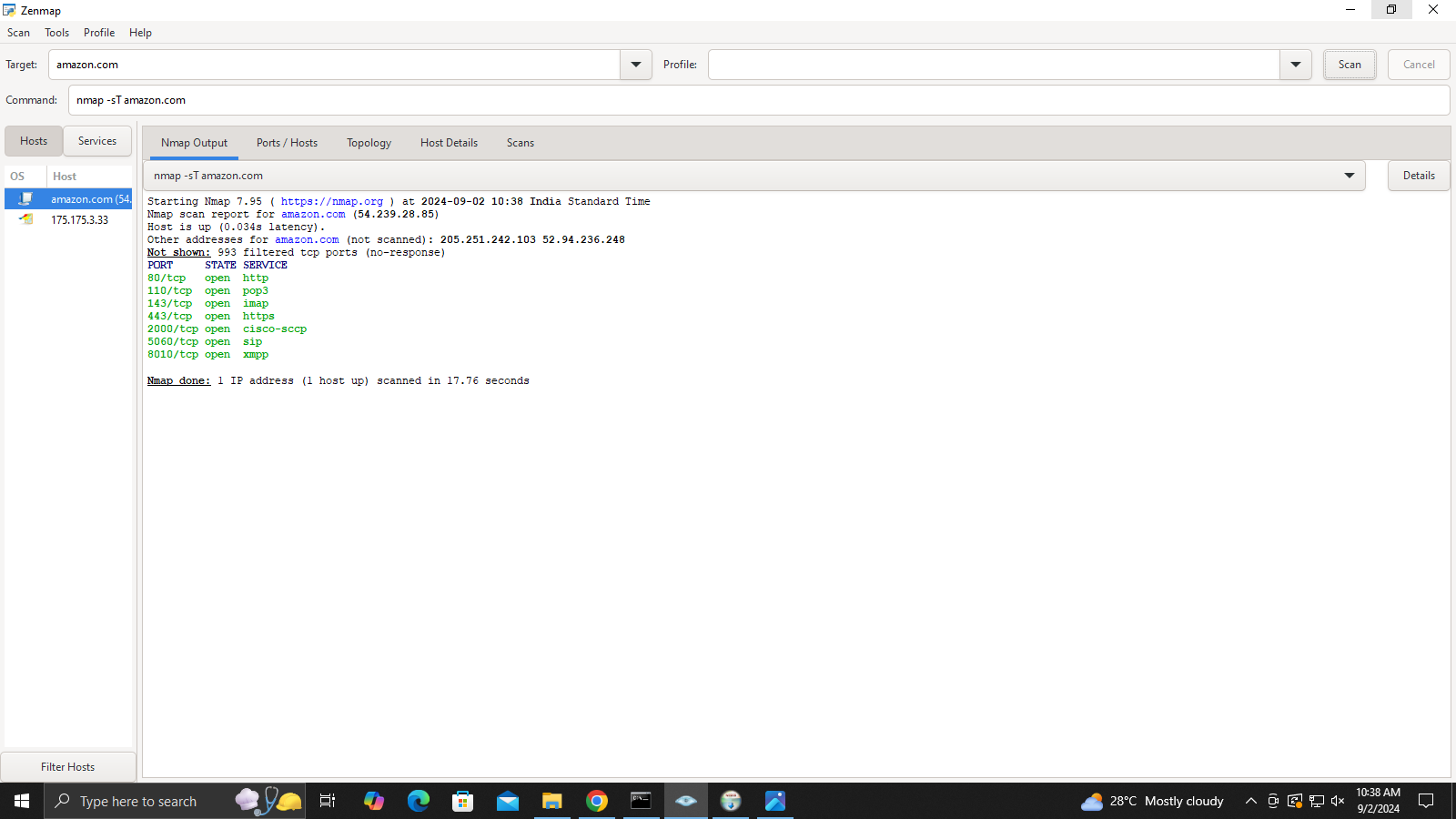
1. **To perform OS fingerprinting**

****

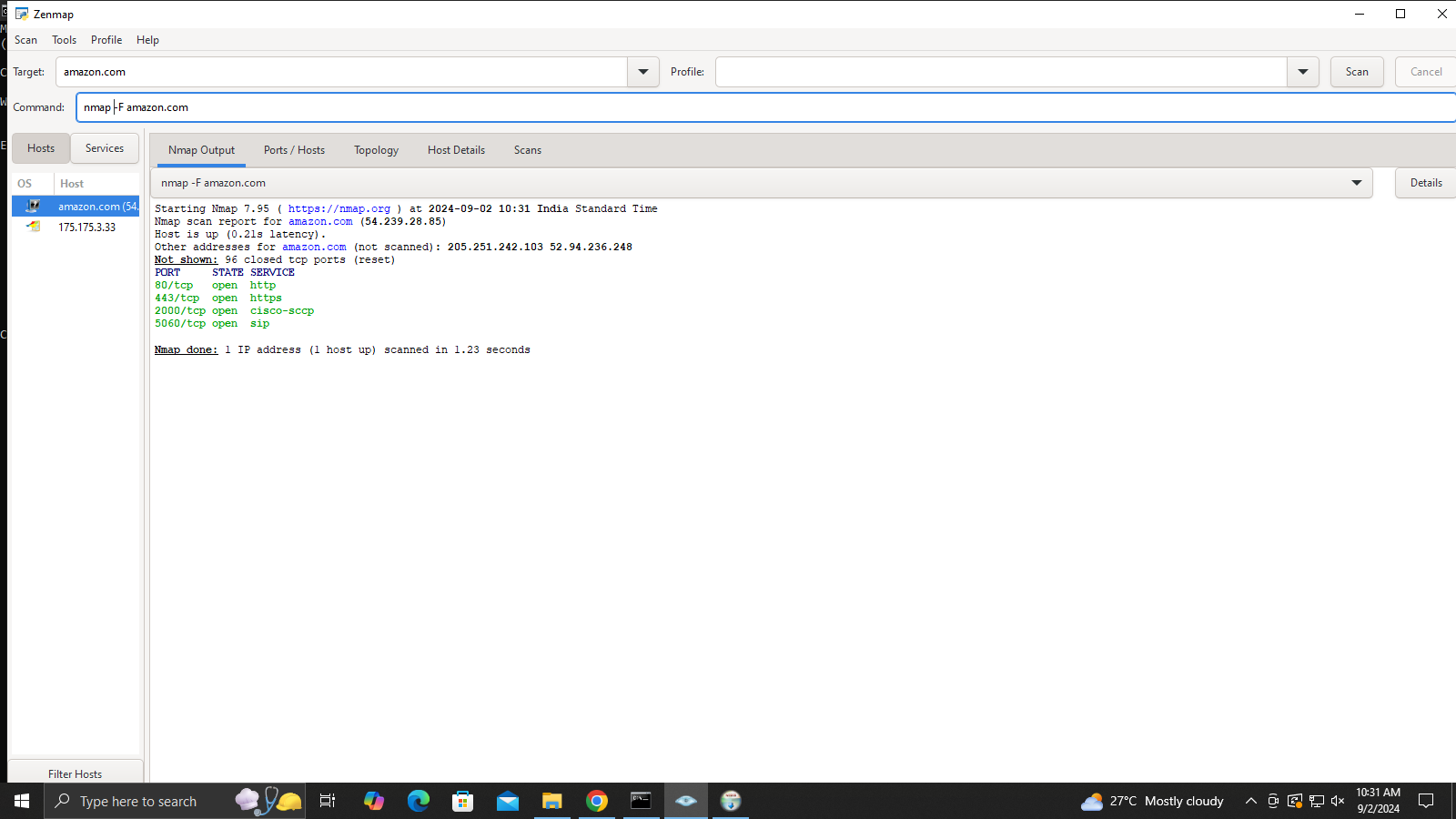
1. **To perform a ping scan**

****

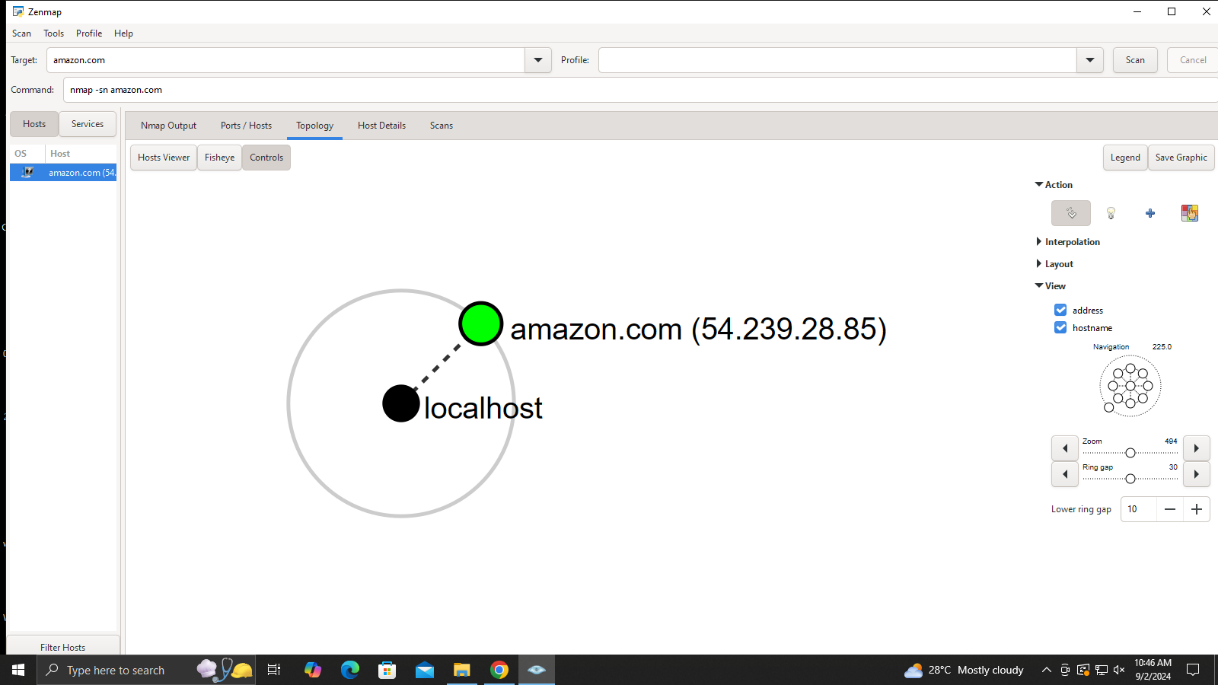
1. **To perform a TCP port scan**

****

1. **To perform a fast scan**

****

1. **Topology of ping scan**

****