# Task 1 — Local Network Port Scan

## Project Overview

**Objective:** Scan your local network to discover open ports and services, analyze potential security risks, and prepare a short report for submission.

## Tools

* **Nmap** — network scanner (https://nmap.org)
* Terminal / PowerShell

## Prerequisites

1. Administrative or sudo privileges on your machine (required for SYN scans).
2. Nmap installed and available in PATH.

## Quick Commands

1. Find your local IP address / range

* Windows (PowerShell / CMD):
* ipconfig
* Linux / macOS:
* ifconfig  
  # or  
  ip addr show

1. Basic TCP SYN scan of common ports across the /24 network:

nmap -sS 192.168.1.0/24 -oN scan\_results.txt

1. More detailed scan (service detection, scripts, version detection):

nmap -sS -sV -sC -p- 192.168.1.0/24 -oN scan\_detailed.txt

1. Save XML output (for tools or parsing):

nmap -sS 192.168.1.0/24 -oX scan\_results.xml

## README sections

### 1) Environment

* OS: e.g., Windows 11 / Ubuntu 22.04
* Nmap version: nmap --version
* Local IP range scanned: 192.168.1.0/24 (replace with yours)

### 2) Commands used

* nmap -sS 192.168.1.0/24 -oN scan\_results.txt
* nmap -sS -sV -sC -p- 192.168.1.0/24 -oN scan\_detailed.txt

### 3) Key Findings (from scan\_results.txt)

### **PORT STATE SERVICE**

### 135/tcp open msrpc

### 139/tcp open netbios-ssn

### 445/tcp open microsoft-ds

### 902/tcp open iss-realsecure

### 912/tcp open apex-mesh

### 1521/tcp open oracle

### 3306/tcp open mysql

### 8080/tcp open http-proxy

### 4) Security Analysis

* **135/tcp — msrpc — Medium → High**
* **Why:** RPC is used by many Windows services for remote procedure calls. Historical vulnerabilities and info disclosure have impacted this service. Exposed RPC increases attack surface (remote code exec historically seen).
* **Risk if exposed:** Remote exploitation, enumeration of services and users.
* **139/tcp & 445/tcp — netbios-ssn / SMB (Microsoft-DS) — High**
* **Why:** SMB enables file sharing and remote admin; it is a frequent target for lateral movement, credential theft, and ransomware. SMBv1 in particular is highly dangerous if enabled.
* **Risk if exposed:** Unauthorized file access, remote code execution, worm/ransomware propagation, credential dumps (via SMB/NTLM abuse).
* **902/tcp & 912/tcp — Management/Agent Ports — Medium**
* **Why:** Ports like 902/912 are often used by virtualization management (VMware) or vendor agents. If default or unauthenticated, they can allow management actions or information disclosure.
* **Risk if exposed:** Remote management abuse, reconnaissance, possible exploitation if software is unpatched.
* **1521/tcp — Oracle listener — High (if default/unpatched)**
* **Why:** Database listeners can allow unauthorized query execution or be used to pivot. Many Oracle vulnerabilities historically target the listener or database engine.
* **Risk if exposed:** Data exfiltration, SQL injection exploitation surface, privileged access if credentials are weak.
* **3306/tcp — MySQL — High (if accessible & not bound to localhost)**
* **Why:** Direct DB access over network can lead to data leaks and privilege escalation if credentials are weak or remote root is allowed.
* **Risk if exposed:** Data theft, account compromise**.**
* **8080/tcp — HTTP (alternate) — Medium**
* **Why:** Web interfaces frequently have vulnerabilities (misconfigurations, outdated frameworks, default creds). If this is a management UI, risk increases.
* **Risk if exposed:** Remote code execution via vulnerable web apps, credential compromise, info disclosure.

### 5) Remediation Steps

**A. Short-term (immediate)**

1. **Restrict access via firewall / host-based firewall**
   * Block incoming access from untrusted interfaces to the listed ports. Example ufw (Linux):
   * sudo ufw deny proto tcp from any to 192.168.1.1 port 135
   * sudo ufw deny proto tcp from any to 192.168.1.1 port 139
   * sudo ufw deny proto tcp from any to 192.168.1.1 port 445
   * sudo ufw deny proto tcp from any to 192.168.1.1 port 1521
   * sudo ufw deny proto tcp from any to 192.168.1.1 port 3306
   * sudo ufw deny proto tcp from any to 192.168.1.1 port 8080
   * On Windows: configure Windows Defender Firewall inbound rules to restrict SMB/RPC to only trusted IPs.
2. **If services aren’t required, stop them**
   * Example (Linux): sudo systemctl stop mysql / sudo systemctl disable mysql — but only if those services are not needed.
3. **Change default or weak credentials**
   * Ensure DB users (Oracle/MySQL) and any web admin accounts have strong passwords and that default accounts are disabled.

**B. Medium-term**

1. **Enable encryption and strong authentication**
   * Use TLS for database connections where supported. For web UI, enable HTTPS and HSTS.
2. **Patch & update software**
   * Update OS packages, Oracle/MySQL server versions, vendor management agents, and any web apps on 8080. Apply vendor security patches.
3. **Disable SMBv1** (if applicable) and harden SMB settings.
   * On Windows: disable SMBv1 feature. Configure SMB signing and limit shares accessible via network.
4. **Network segmentation / ACLs**
   * Place critical services (DBs, management) on management VLANs accessible only from admin subnets.

**C. Longer-term & verification**

1. **Run targeted enumeration & version detection**
   * nmap -sV -p 135,139,445,902,912,1521,3306,8080 192.168.56.1 -oN scan\_with\_versions.txt
   * nmap -A -p- 192.168.1.1 (careful — noisy)  
     These will identify exact software versions to check for known CVEs.
2. **Credential audit and access control**
   * Rotate all DB and management credentials. Enforce least privilege.
3. **Vulnerability scan and patch management**
   * Use OpenVAS, Nessus, or Nexpose (or vendor tools) to search for known vulnerabilities against the discovered services.
4. **Logging, monitoring & IDS**
   * Ensure logs (system, DB, application) are collected and monitored. Consider deploying host-based intrusion detection (OSSEC) or network IDS (Snort/Suricata).
5. **Penetration testing (authorized)**
   * After hardening, run an authorized pen-test to verify controls.

### 6) Screenshot

