“Securing Ports Using Nmap – Zenmap GUI”

Editor – Arya Mokashi

Project Overview - “Securing Ports Using Nmap – Zenmap GUI”

Objective

To explore port scanning with Nmap – Zenmap GUI, identify open/filtered/closed ports, and secure vulnerable ports (specifically MySQL port 3306) to minimize unauthorized access risks.

Hands-on Learning Project
I recently explored how network ports work, how to identify active services, and how to improve local security using tools like Zenmap (Nmap GUI) and system configurations.

Steps Performed

1. **Port Scanning with Zenmap**

   Ran an intense scan on my IPv4 address using Zenmap GUI.

   Identified open, filtered, and closed ports.

2. **Vulnerability Identification**

   Found **Port 3306 (MySQL)** open and accessible over the network:

   3306/tcp open mysql (unauthorized)

3. **MySQL Security Verification**

   Checked MySQL user security.

   Verified that root@localhost required a password, preventing direct unauthorized login.

4. **Configuration Hardening**

   Restricted MySQL to local connections by editing:

   C:\ProgramData\MySQL\MySQL Server 8.0\my.ini

   ```
   [mysqld]

   bind-address = 127.0.0.1
   ```

5. Restarted

   ```
   net stop MySQL80
   ```

net start MySQL80

6. **Firewall Hardening**
   Blocked remote access to port 3306 with Windows Firewall rule:
   netsh advfirewall firewall add rule name="Block MySQL Remote Access" dir=in action=block protocol=TCP localport=3306

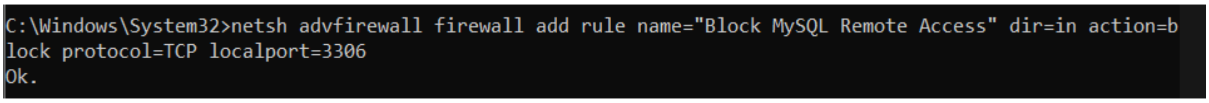
7. **Verification**
   - Rescanned with Nmap.
   - Port 3306 appeared as **unauthorized/filtered**, confirming that external access was blocked.

Key Learnings

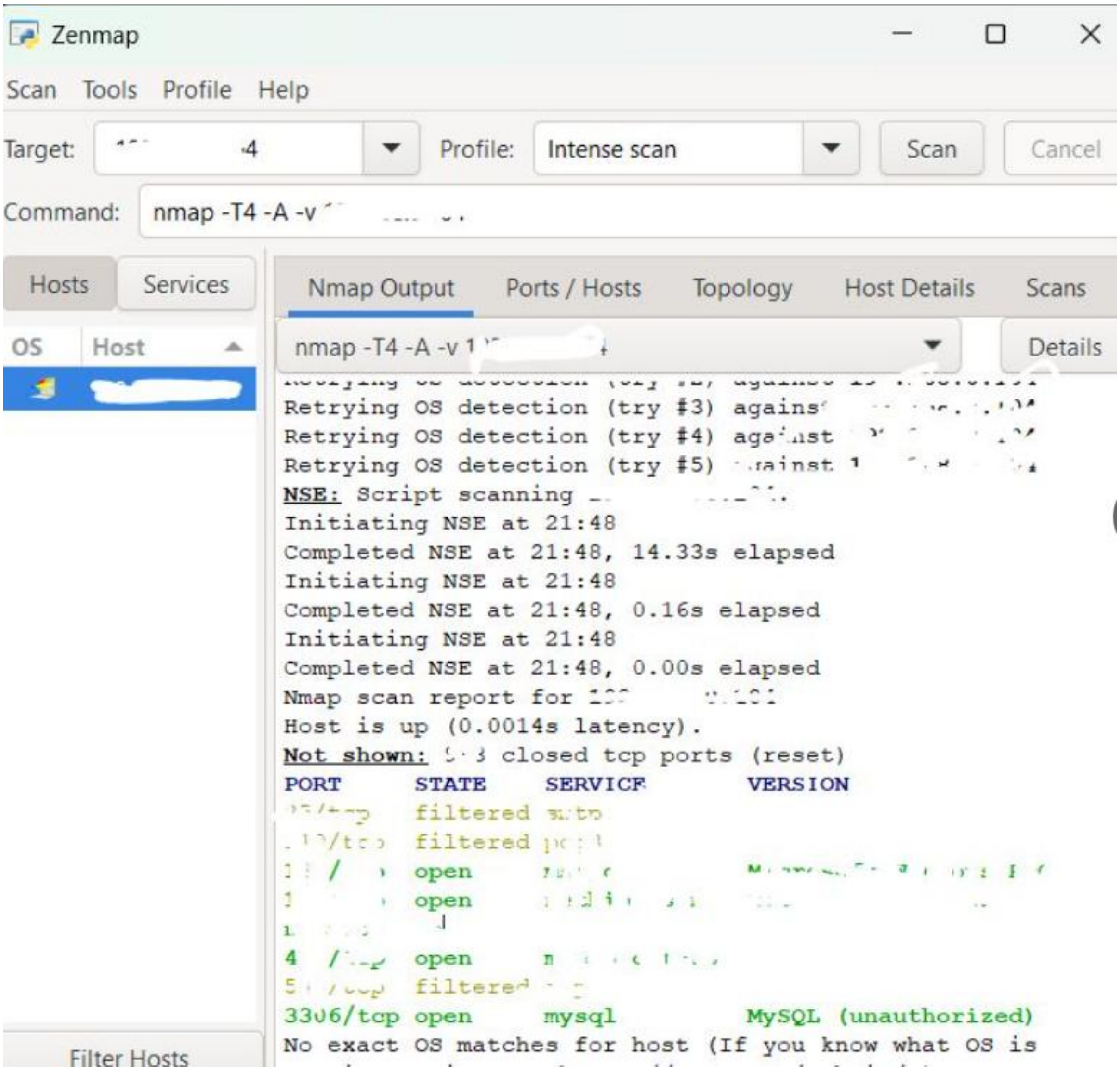
- **Ports and Security**
  - Ports act as logical endpoints at the **transport layer (Layer 4)**, directing data to the right application (e.g., HTTP: 80, HTTPS: 443).
  - Open ports = potential attack surface.
  - Filtered ports = protected/hidden by firewall rules.
  - Closed ports = no active service but reachable.
- **Security Best Practices**
  - Services not in use should be **closed**.
  - Critical services (like MySQL) should be **restricted to localhost** unless remote access is explicitly needed.
  - Strong authentication (passwords, role-based access) is crucial.
  - Firewalls should block unnecessary exposure of ports to the network.

Conclusion

Port scanning with Zenmap revealed that MySQL (port 3306) was network-visible and posed a potential risk. By reconfiguring MySQL to allow only local connections and blocking the port through the Windows firewall, the risk was mitigated. Filtered (unauthorized) ports do not necessarily indicate a vulnerability, but **open ports without proper authentication are high-risk**. Effective security requires minimizing the attack surface by keeping only essential ports open and well-protected.



```
C:\Windows\System32>netsh advfirewall firewall add rule name="Block MySQL Remote Access" dir=in action=b
lock protocol=TCP localport=3306
Ok.
```



Target: [redacted] 4

Profile: Intense scan

Command: nmap -T4 -A -v [redacted]

nmap -T4 -A -v 1[redacted]

[redacted]

```
Retrying OS detection (try #3) against [redacted]
Retrying OS detection (try #4) against [redacted]
Retrying OS detection (try #5) against [redacted]
NSE: Script scanning [redacted]
Initiating NSE at 21:48
Completed NSE at 21:48, 14.33s elapsed
Initiating NSE at 21:48
Completed NSE at 21:48, 0.16s elapsed
Initiating NSE at 21:48
Completed NSE at 21:48, 0.00s elapsed
Nmap scan report for [redacted]
Host is up (0.0014s latency).
Not shown: [redacted] closed tcp ports (reset)
PORT     STATE    SERVICE       VERSION
[redacted]/tcp filtered [redacted]
[redacted]/tcp filtered [redacted]
[redacted]     open     [redacted]
[redacted]     open     [redacted]
[redacted]
4[redacted]    open     [redacted]
5[redacted]    filtered [redacted]
3306/tcp open     mysql         MySQL (unauthorized)
No exact OS matches for host (If you know what OS is
```