Vulnerability Scanning & Hardening Lab

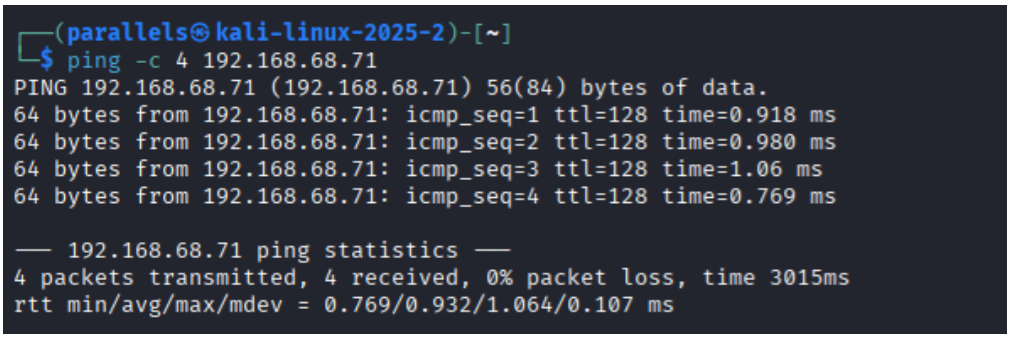
Performed by: Jorden Plaines

Date: October 2025

Tools Used: Windows 11, Kali Linux, Nmap, PowerShell, Windows Firewall

Environment: Parallels Desktop on macOS (Simulated Attacker/Target Lab)

Step 1 — Network Verification (Kali)



Purpose:

Verify that the Windows 11 target (192.168.68.71) is reachable over the network before performing scans.

Outcome:

All ping requests were successful with 0% packet loss, confirming that the target host is active and communicating on the network.

Step 2 — Host Discovery (Kali)

Figure 2 — Nmap host discovery from Kali identifying live devices on the 192.168.68.0/24 network, confirming visibility of the Windows target (192.168.68.71).

Purpose:

To identify all active devices on the local subnet (192.168.68.0/24) before performing targeted scans. This ensures that the correct host (192.168.68.71) can be isolated for deeper analysis.

Outcome:

Nmap successfully discovered multiple devices across the network, including smart home devices, Apple systems, and the Windows 11 target (192.168.68.71). The scan detected 22 active hosts out of 256 possible IPs, confirming that the target is reachable and active within the subnet.

Step 3 — Port Scanning & Service Enumeration (Kali)

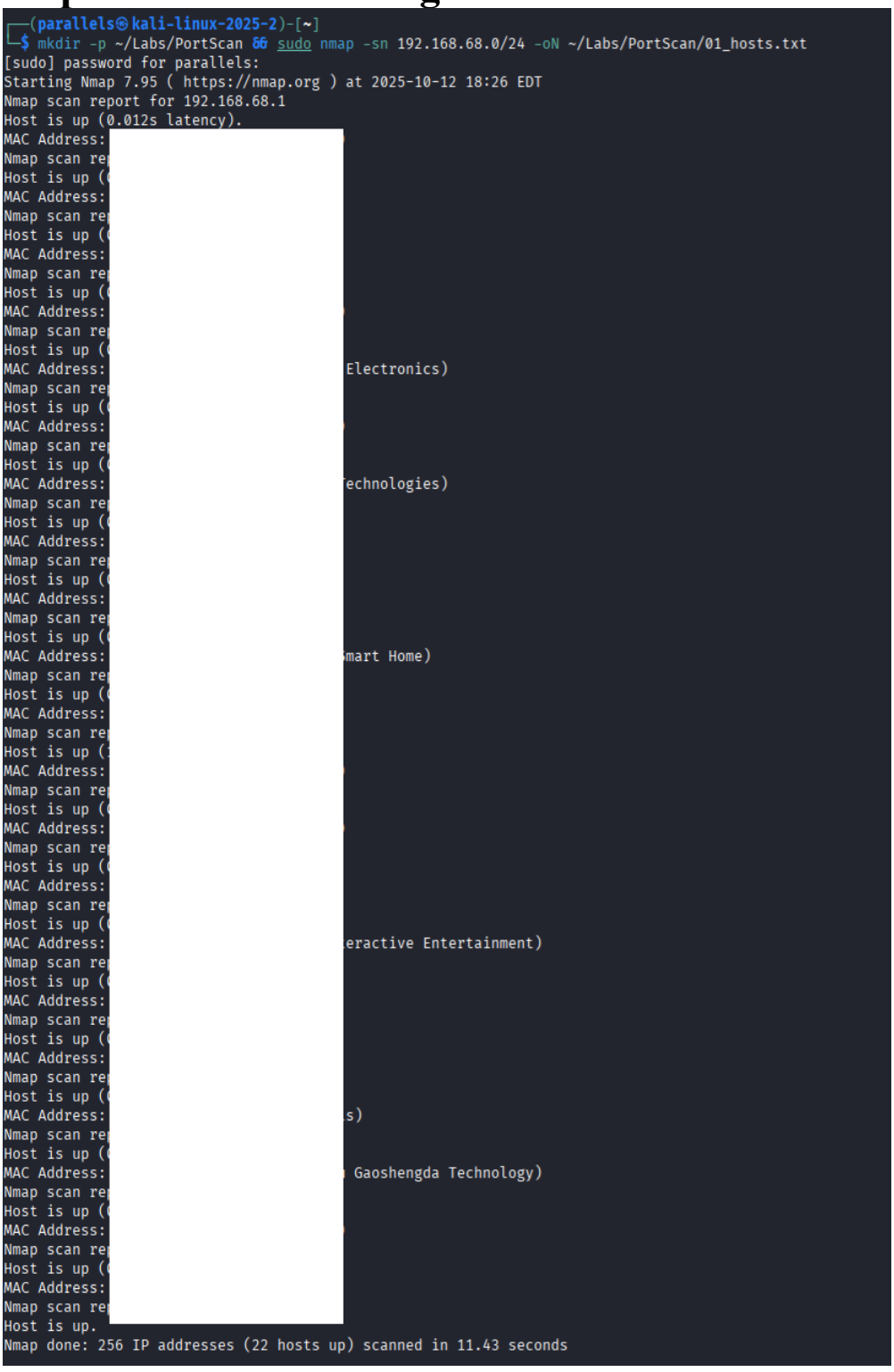


Figure 3 — Nmap port scans from Kali. Broad scan indicates multiple common service ports are filtered (possible firewall). Targeted scans will follow to identify actual service versions and owning processes.

Purpose:

Identify open or filtered TCP ports on the Windows target to determine which services are reachable and worth deeper enumeration.

Outcome / Interpretation:

- The host responded to port scans and many well-known service ports (22, 80, 135, 139, 445, 3389, 5985, 5986, etc.) are shown as **filtered** in the broad scan — this typically indicates a firewall or packet filtering is present.
- Filtered ports mean probe packets were dropped or blocked; it doesn't prove the service isn't running, only that it's not reachable from the scanner without further enumeration or credentialed access.
- These results justify the next steps: targeted service/version scans and mapping ports to Windows processes to confirm which services are listening.

Step 4 — Host Verification & Listening Ports (Windows)

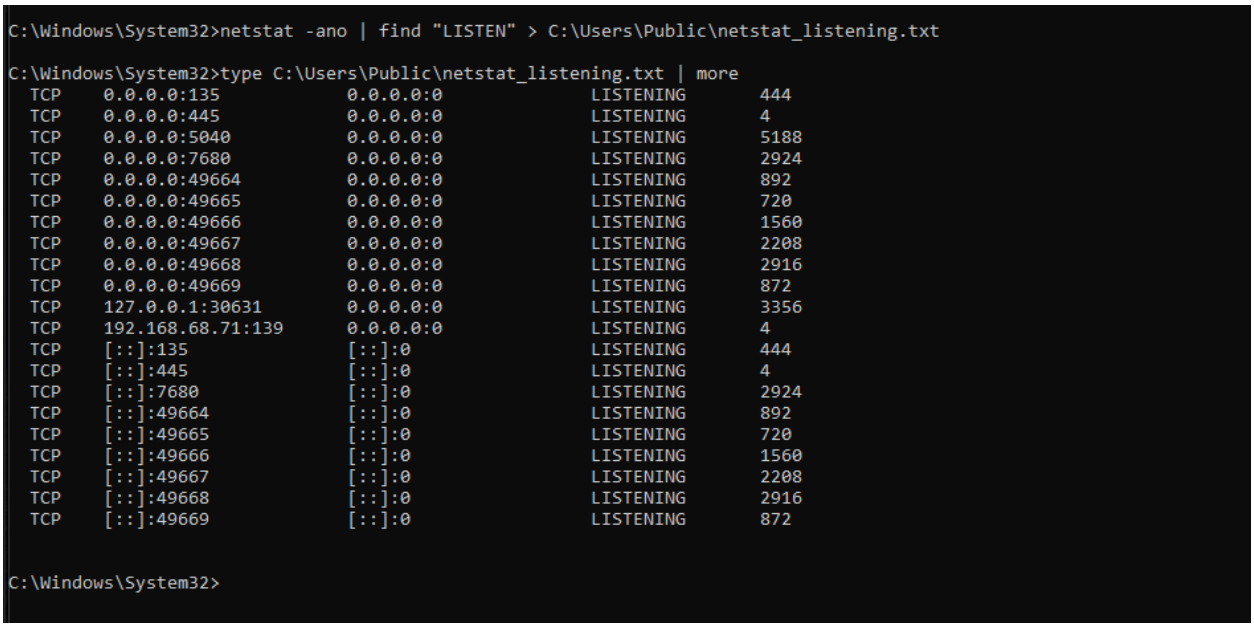


Figure 4 — netstat output on the Windows target listing listening TCP ports and associated PIDs (used to map ports → processes).

Purpose:

Confirm which TCP ports are actually listening on the Windows target and capture the owning PIDs (so we can map services to processes in a later step).

Outcome / Interpretation:

The netstat output shows the system is listening on several ports including **135, 445, 5040, 7680** and many ephemeral ports. The right-most column lists process IDs (PIDs) for each listener — this proves the services are locally bound and gives you the IDs to correlate to process names (using Get-Process -Id <pid>). This is the evidence needed to show which services exposed SMB/RPC prior to hardening.

Step 5 — Firewall Analysis & Temporary SMB Test (Windows + Kali)

Purpose

Audit and document the current Windows firewall configuration, then create a temporary inbound rule to allow SMB traffic (TCP 445) for testing.

This verifies that the firewall is responsible for blocking SMB connections and demonstrates how enabling the port affects external reachability.

5.1 Firewall Rule Enumeration (Windows)

Exported and reviewed all firewall rules to establish a baseline before making any changes.

The command output lists many built-in and application-specific rules (e.g., Windows Media Player, Camera, Network Discovery).

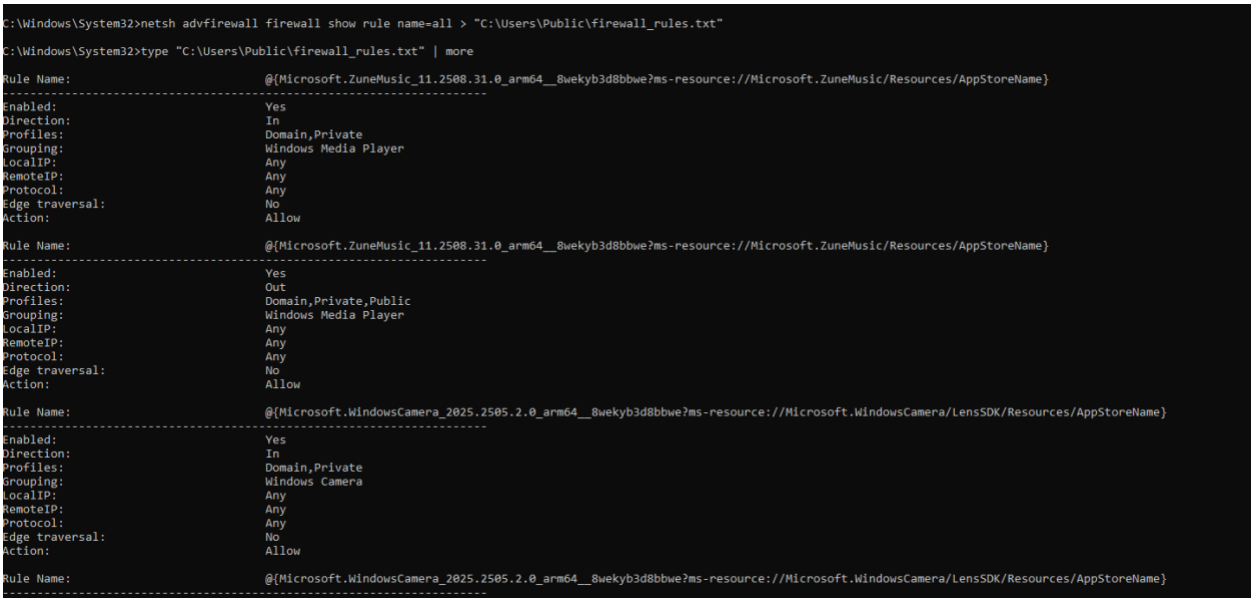
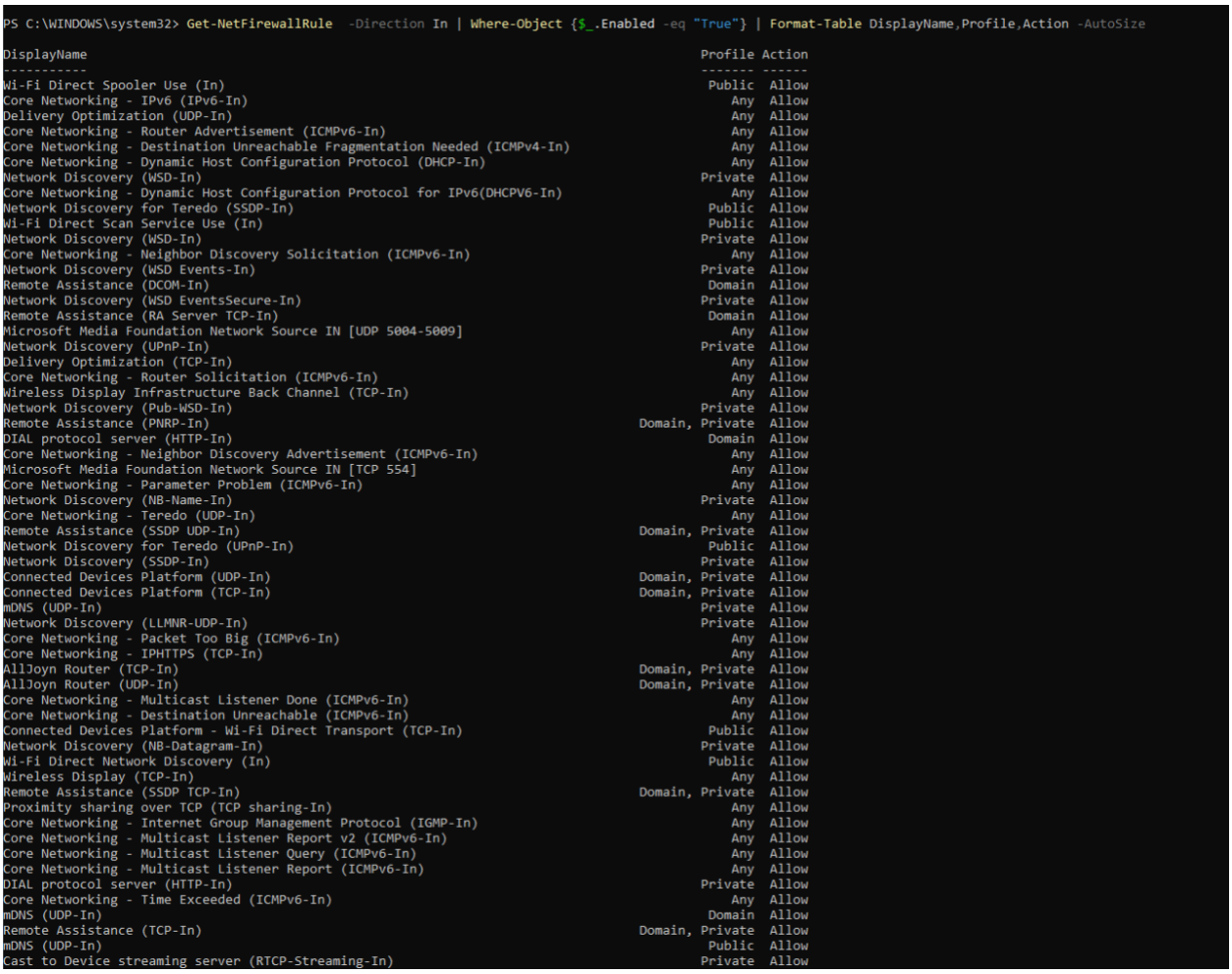


Figure 5.1 — Exported firewall rules (baseline) showing enabled inbound and outbound rules for system apps prior to modification.

5.2 Enabled Inbound Rules (Windows)

Filtered the ruleset to display only active inbound rules.

This view highlights services such as Network Discovery, Remote Assistance, and Core Networking, which define how the host communicates within its local domain.



```
PS C:\WINDOWS\system32> Get-NetFirewallRule  -Direction In | Where-Object {$_.Enabled -eq "True"} | Format-Table DisplayName,Profile,Action -AutoSize

DisplayName                                                                     Profile Action
-----------                                                                     ------- ------
Wi-Fi Direct Spooler Use (In)                                                    Public  Allow
Core Networking - IPv6 (IPv6-In)                                                    Any  Allow
Delivery Optimization (UDP-In)                                                      Any  Allow
Core Networking - Router Advertisement (ICMPv6-In)                                  Any  Allow
Core Networking - Destination Unreachable Fragmentation Needed (ICMPv4-In)          Any  Allow
Core Networking - Dynamic Host Configuration Protocol (DHCP-In)                     Any  Allow
Network Discovery (WSD-In)                                                      Private  Allow
Core Networking - Dynamic Host Configuration Protocol for IPv6(DHCPV6-In)            Any  Allow
Network Discovery for Teredo (SSDP-In)                                           Public  Allow
Wi-Fi Direct Scan Service Use (In)                                               Public  Allow
Network Discovery (WSD-In)                                                      Private  Allow
Core Networking - Neighbor Discovery Solicitation (ICMPv6-In)                       Any  Allow
Network Discovery (WSD Events-In)                                               Private  Allow
Remote Assistance (DCOM-In)                                                      Domain  Allow
Network Discovery (WSD EventsSecure-In)                                         Private  Allow
Remote Assistance (RA Server TCP-In)                                             Domain  Allow
Microsoft Media Foundation Network Source IN [UDP 5004-5009]                        Any  Allow
Network Discovery (UPnP-In)                                                     Private  Allow
Delivery Optimization (TCP-In)                                                      Any  Allow
Core Networking - Router Solicitation (ICMPv6-In)                                   Any  Allow
Wireless Display Infrastructure Back Channel (TCP-In)                               Any  Allow
Network Discovery (Pub-WSD-In)                                                  Private  Allow
Remote Assistance (PNRP-In)                                             Domain, Private  Allow
DIAL protocol server (HTTP-In)                                                   Domain  Allow
Core Networking - Neighbor Discovery Advertisement (ICMPv6-In)                      Any  Allow
Microsoft Media Foundation Network Source IN [TCP 554]                              Any  Allow
Core Networking - Parameter Problem (ICMPv6-In)                                     Any  Allow
Network Discovery (NB-Name-In)                                                  Private  Allow
Core Networking - Teredo (UDP-In)                                                   Any  Allow
Remote Assistance (SSDP UDP-In)                                         Domain, Private  Allow
Network Discovery for Teredo (UPnP-In)                                           Public  Allow
Network Discovery (SSDP-In)                                                     Private  Allow
Connected Devices Platform (UDP-In)                                     Domain, Private  Allow
Connected Devices Platform (TCP-In)                                     Domain, Private  Allow
mDNS (UDP-In)                                                                   Private  Allow
Network Discovery (LLMNR-UDP-In)                                                Private  Allow
Core Networking - Packet Too Big (ICMPv6-In)                                        Any  Allow
Core Networking - IPHTTPS (TCP-In)                                                  Any  Allow
AllJoyn Router (TCP-In)                                                 Domain, Private  Allow
AllJoyn Router (UDP-In)                                                 Domain, Private  Allow
Core Networking - Multicast Listener Done (ICMPv6-In)                               Any  Allow
Core Networking - Destination Unreachable (ICMPv6-In)                               Any  Allow
Connected Devices Platform - Wi-Fi Direct Transport (TCP-In)                     Public  Allow
Network Discovery (NB-Datagram-In)                                              Private  Allow
Wi-Fi Direct Network Discovery (In)                                              Public  Allow
Wireless Display (TCP-In)                                                           Any  Allow
Remote Assistance (SSDP TCP-In)                                         Domain, Private  Allow
Proximity sharing over TCP (TCP sharing-In)                                         Any  Allow
Core Networking - Internet Group Management Protocol (IGMP-In)                      Any  Allow
Core Networking - Multicast Listener Report v2 (ICMPv6-In)                          Any  Allow
Core Networking - Multicast Listener Query (ICMPv6-In)                              Any  Allow
Core Networking - Multicast Listener Report (ICMPv6-In)                             Any  Allow
DIAL protocol server (HTTP-In)                                                  Private  Allow
Core Networking - Time Exceeded (ICMPv6-In)                                         Any  Allow
mDNS (UDP-In)                                                                    Domain  Allow
Remote Assistance (TCP-In)                                              Domain, Private  Allow
mDNS (UDP-In)                                                                    Public  Allow
Cast to Device streaming server (RTCP-Streaming-In)                             Private  Allow
```

Figure 5.2 — Filtered list of active inbound firewall rules showing which profiles (Domain, Private, Public) currently allow inbound connections.

5.3 Create Temporary SMB Rule (Windows)

A temporary inbound rule named **Allow SMB Test** was added to permit inbound TCP traffic on port 445.

This provides a controlled way to validate whether the port becomes reachable externally.

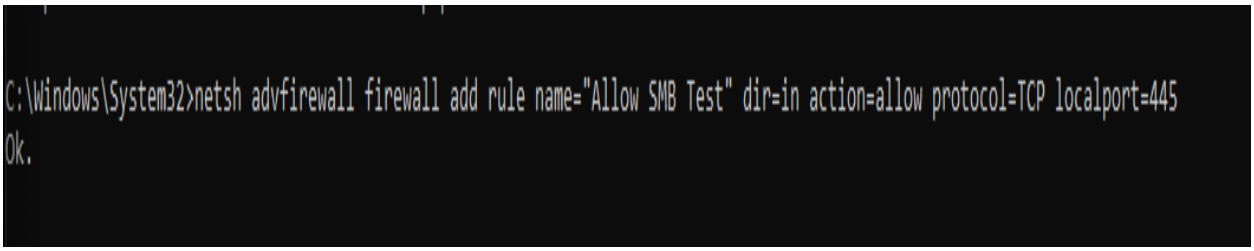


Figure 5.3 — Temporary inbound rule "Allow SMB Test" created to open TCP 445 for controlled testing.

5.4 Validate from Kali (Linux Attacker Machine)

From the Kali VM, a targeted Nmap scan was performed against 192.168.68.71 on port 445.

The scan result confirmed that the port was **open**, proving the firewall rule successfully allowed external SMB traffic.

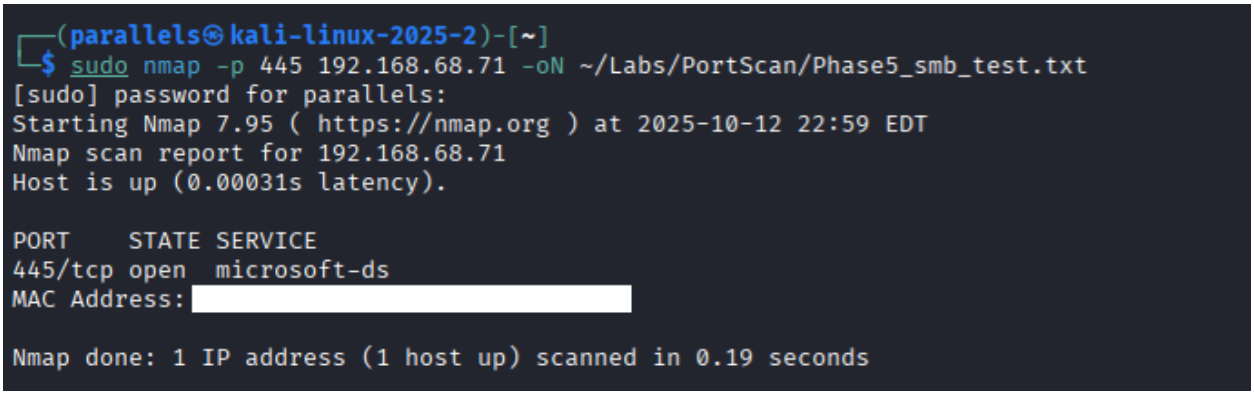


Figure 5.4 — Nmap scan from Kali showing port 445 open (microsoft-ds service) when the temporary Allow SMB rule is active.

5.5 Delete Temporary Rule (Windows)

After validation, the “Allow SMB Test” rule was deleted to restore the system’s secure configuration.

This returns the firewall to its original hardened state where SMB is blocked from external hosts.

```
C:\Windows\System32>netsh advfirewall firewall delete rule name="Allow SMB Test"

Deleted 1 rule(s).
Ok.
```

Figure 5.5 — Firewall rule successfully deleted after testing, restoring the default security posture.

Summary / Key Findings

- Confirmed that the Windows firewall is the primary control restricting SMB access.
- Verified the effect of enabling/disabling port 445 from an external attacker viewpoint.
- Demonstrated safe, reversible configuration changes as part of the hardening workflow.

Step 6 — Service and Process Verification (Windows + Kali)

Purpose

Identify which services are actively running on the target system and map them to their corresponding process IDs (PIDs).

This step bridges network-level visibility from Nmap with host-level visibility inside Windows, confirming which processes are listening on the open ports discovered earlier.

6.1 Service Detection and Version Scan (Kali)

From Kali, an Nmap service/version scan was run against the target to identify open and filtered ports.

The scan detected **port 7680 (pando-pub)** as open, while common ports like 135, 139, 445, and 5040 were filtered — confirming the firewall and SMB restrictions from earlier steps are working as intended.

```
└─$ sudo nmap -sV -sC -p 135,139,445,5040,7680 192.168.68.71 -oN ~/Labs/PortScan/03_service_version_common.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-12 23:04 EDT
Nmap scan report for 192.168.68.71
Host is up (0.00031s latency).

PORT     STATE    SERVICE      VERSION
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
445/tcp  filtered microsoft-ds
5040/tcp filtered unknown
7680/tcp open     pando-pub?
MAC Address: [redacted]

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.68 seconds
```

Figure 6.1 — Nmap service/version scan results showing port 7680 open and others filtered due to firewall restrictions.

6.2 Process Identification (Windows)

Using PowerShell, the open ports identified from the Nmap results were correlated to running Windows processes via their **PIDs**.

The output shows each listening port corresponds to system services such as **svchost**, **services**, and **prl_tools_service** (a Parallels component).

This confirms which processes own the network sockets previously observed in the Nmap scan.

```
PS C:\WINDOWS\system32> Get-Process -Id 444,4,5188,2924,872,3356 | Format-Table ID,ProcessName -AutoSize

  Id ProcessName
  -- -----------
3356 prl_tools_service
 872 services
 444 svchost
2924 svchost
5188 svchost
   4 System
```

Figure 6.2 — PowerShell output mapping listening ports to specific Windows processes, verifying which services are actively bound to network interfaces.

Summary / Key Findings

- Confirmed service enumeration aligns with previous scan results.
- Validated that the open port (7680) belongs to a legitimate Windows service.
- Demonstrated host-level verification of network exposure, correlating external scan data with local process information.

Step 7 — Vulnerability Scan (Kali)

Purpose

Conduct an Nmap vulnerability scan to identify potential weaknesses or outdated services on the target system.

This step evaluates whether any common ports (135, 139, 445, 5040) expose known vulnerabilities or misconfigurations before and after system hardening.

7.1 Nmap Vulnerability Script Scan

Using Kali Linux, a targeted **Nmap vulnerability script (--script vuln)** was executed against the Windows host.

The scan focused on SMB-related ports and common service ports associated with Windows systems.

The results show that all scanned ports (135, 139, 445, 5040) were **closed**, indicating that the system's firewall and SMB hardening measures were successfully preventing external access.

```
┌──(parallels㉿kali-linux-2025-2)-[~]
└─$ sudo nmap --script vuln -p 135,139,445,5040 192.168.68.71 -oN ~/Labs/VulnScan/phase7_vulnscan.txt
[sudo] password for parallels:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-13 15:30 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for 192.168.68.71
Host is up (0.51s latency).

PORT     STATE  SERVICE
135/tcp  closed msrpc
139/tcp  closed netbios-ssn
445/tcp  closed microsoft-ds
5040/tcp closed unknown
MAC Address: F6:F9:B5:82:F5:67 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 38.06 seconds
```

Figure 7.1 — Nmap vulnerability scan output confirming that all major Windows service ports are closed and no vulnerabilities are exposed externally.

Summary / Key Findings

- Verified that previously open or filtered ports are now closed post-hardening.
- No active vulnerabilities were detected by the Nmap vulnerability script.
- Confirms effective implementation of prior security measures, including firewall restrictions and SMB protocol hardening.

Step 8 — SMB and Firewall Hardening (Windows + Kali)

Purpose

Enhance the security posture of the Windows host by disabling outdated SMB protocols, enforcing SMB signing, and implementing strict inbound firewall rules.

These steps directly mitigate lateral movement risks, unauthorized network access, and legacy vulnerabilities like EternalBlue (MS17-010).

8.1 Baseline Firewall Export (Windows)

Before applying any changes, the existing firewall configuration was exported.

This provided a snapshot of the pre-hardening state and served as a rollback reference if required.

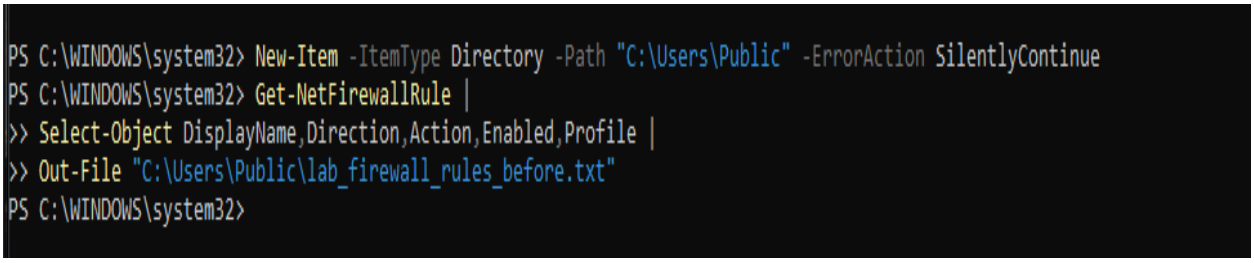


Figure 8.1 — Export of baseline Windows firewall rules prior to making SMB and port configuration changes.

8.2 SMB Protocol Hardening (Windows)

Disable SMBv1 Protocol

SMBv1 is a legacy file-sharing protocol with known vulnerabilities.

The command output confirms both **SMB1Protocol** and **SMB1Protocol-Server** were disabled system-wide.

```
PS C:\WINDOWS\system32> Get-WindowsOptionalFeature -Online -FeatureName SMB1Protocol-Server


FeatureName      : SMB1Protocol-Server
DisplayName      : SMB 1.0/CIFS Server
Description      : Support for the SMB 1.0/CIFS file server for sharing data with legacy clients and browsing the network neighborhood.
RestartRequired  : Possible
State            : Disabled
CustomProperties :
                   ServerComponent\Description : Support for the SMB 1.0/CIFS file server for sharing data with legacy clients and browsing the network neighborhood.
                   ServerComponent\DisplayName : SMB 1.0/CIFS Server
                   ServerComponent\Id : 1035
                   ServerComponent\InstallWithParentByDefault : true
                   ServerComponent\Parent : FS-SMB1
                   ServerComponent\Type : Feature
                   ServerComponent\UniqueName : FS-SMB1-SERVER
                   ServerComponent\Deploys\Update\Name : SMB1Protocol-Server



PS C:\WINDOWS\system32> Disable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol        -NoRestart


Path          :
Online        : True
RestartNeeded : False
```

Figure 8.2a — Verification showing SMBv1 features disabled, preventing insecure legacy connections.

Disable SMBv1 Feature and Add Firewall Rule

Re-checked the SMB1 feature status to confirm it remained disabled.

Then, created a new inbound firewall rule named **"Block SMB inbound (Lab Harden)"** to explicitly block TCP 445 across all profiles, further preventing any inbound SMB attempts.

```
PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "Block SMB inboud (Lab Harden)" -Direction Inbound -LocalPort 445 -Protocol TCP -Action Block -Profile Any


Name                            : {20871eff-5ec0-47ea-9aa9-ba1a58c3e623}
DisplayName                     : Block SMB inboud (Lab Harden)
Description                     :
DisplayGroup                    :
Group                           :
Enabled                         : True
Profile                         : Any
Platform                        : {}
Direction                       : Inbound
Action                          : Block
EdgeTraversalPolicy             : Block
LooseSourceMapping              : False
LocalOnlyMapping                : False
Owner                           :
PrimaryStatus                   : OK
Status                          : The rule was parsed successfully from the store. (65536)
EnforcementStatus               : NotApplicable
PolicyStoreSource               : PersistentStore
PolicyStoreSourceType           : Local
RemoteDynamicKeywordAddresses   : {}
PolicyAppId                     :
PackageFamilyName               :


PS C:\WINDOWS\system32> Set-SmbServerConfiguration -RequireSecuritySignature $true -Force
```

Enable SMB Signing (Server and Client)

Enabling SMB signing ensures message integrity and prevents tampering during SMB communication.

Commands were executed to enforce **RequireSecuritySignature = True** for both the SMB **Server** and **Client** configurations.

```
PS C:\WINDOWS\system32> New-NetFirewallRule -DisplayName "Block SMB inboud (Lab Harden)" -Direction Inbound -LocalPort 445 -Protocol TCP -Action Block -Profile Any


Name                            : {20871eff-5ec0-47ea-9aa9-ba1a58c3e623}
DisplayName                     : Block SMB inboud (Lab Harden)
Description                     :
DisplayGroup                    :
Group                           :
Enabled                         : True
Profile                         : Any
Platform                        : {}
Direction                       : Inbound
Action                          : Block
EdgeTraversalPolicy             : Block
LooseSourceMapping              : False
LocalOnlyMapping                : False
Owner                           :
PrimaryStatus                   : OK
Status                          : The rule was parsed successfully from the store. (65536)
EnforcementStatus               : NotApplicable
PolicyStoreSource               : PersistentStore
PolicyStoreSourceType           : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId                     :
PackageFamilyName               :


PS C:\WINDOWS\system32> Set-SmbServerConfiguration -RequireSecuritySignature $true -Force
```

Figure 8.2c — SMB server signing enabled for secure communication validation.

```
PS C:\WINDOWS\system32> Set-SmbClientConfiguration -RequireSecuritySignature $true

Confirm
Are you sure you want to perform this action?
Performing operation 'Modify' on Target 'SMB Client Configuration'.
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): a
PS C:\WINDOWS\system32>
```

Figure 8.2d — SMB client signing enforced to ensure authenticated sessions with servers.

Verify SMB Configuration

Post-modification verification confirmed the server's **SMB1 protocol disabled**, **SMB signing enabled**, and **firewall rule active**.

This validated that all SMB-related mitigations were correctly applied and persistent.

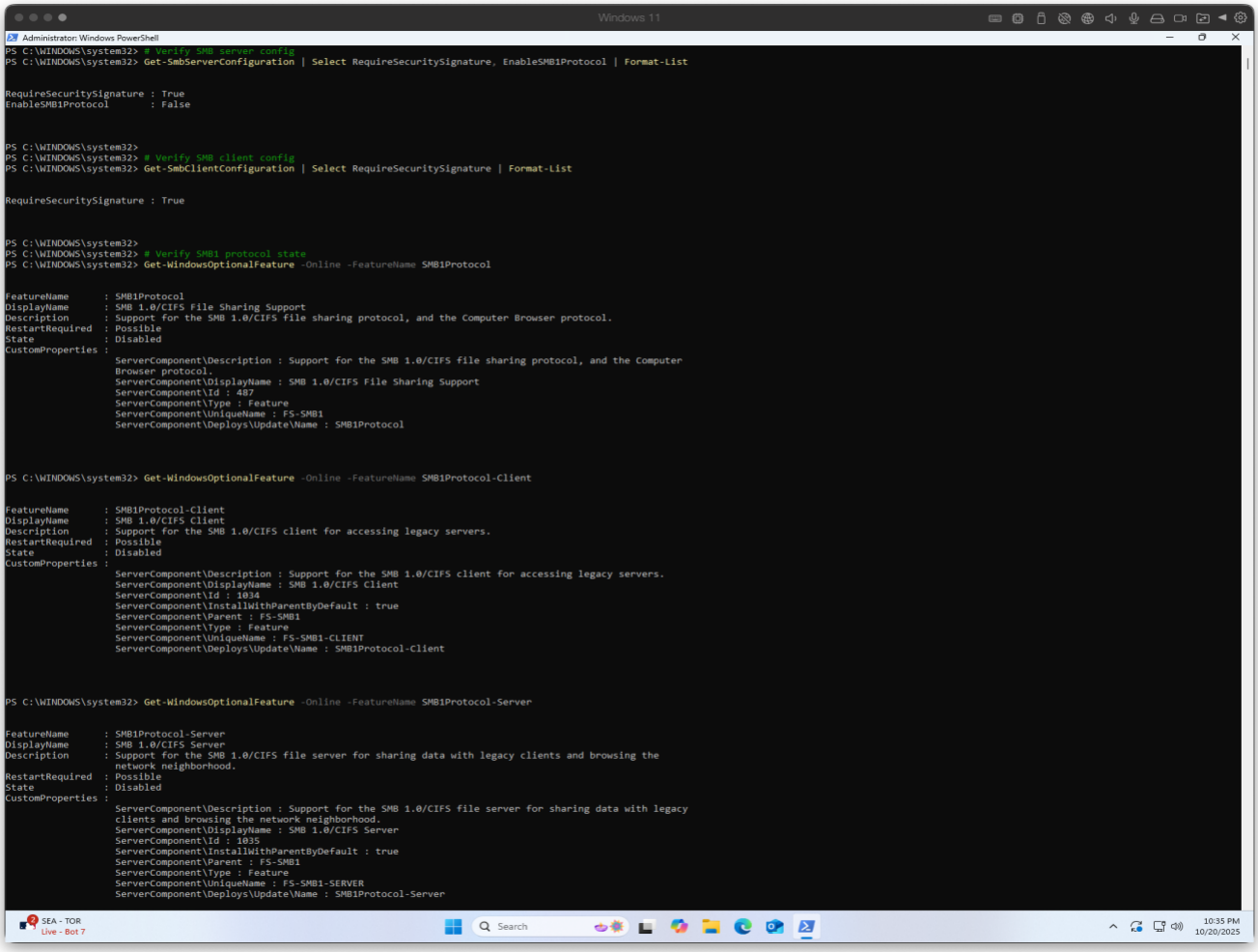


Figure 8.2e — Output confirming SMB hardening: SMBv1 disabled, SMB signing required, and secure configuration applied

Verify Firewall Rule

The firewall rule **"Block SMB inbound (Lab Harden)"** was reviewed to confirm that it successfully blocked all inbound SMB traffic and was stored persistently within the local policy store.

```
PS C:\WINDOWS\system32> Get-NetFirewallRule -DisplayName "Block SMB inboud (Lab Harden)" | Format-List *


Name                          : {20871eff-5ec0-47ea-9aa9-ba1a58c3e623}
ID                            : {20871eff-5ec0-47ea-9aa9-ba1a58c3e623}
DisplayName                   : Block SMB inboud (Lab Harden)
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Block
EdgeTraversalPolicy           : Block
LSM                           : False
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSourceType         : Local
Caption                       :
Description                   :
ElementName                   : Block SMB inboud (Lab Harden)
InstanceID                    : {20871eff-5ec0-47ea-9aa9-ba1a58c3e623}
CommonName                    :
PolicyKeywords                :
PolicyDecisionStrategy        : 2
PolicyRoles                   :
ConditionListType             : 3
CreationClassName             : MSFT|FW|FirewallRule|{20871eff-5ec0-47ea-9aa9-ba1a58c3e623}
ExecutionStrategy             : 2
Mandatory                     :
PolicyRuleName                :
Priority                      :
RuleUsage                     :
SequencedActions              : 3
SystemCreationClassName       :
SystemName                    :
DisplayGroup                  :
LocalOnlyMapping              : False
LooseSourceMapping            : False
Owner                         :
PackageFamilyName             :
Platforms                     : {}
PolicyAppId                   :
PolicyStoreSource             : PersistentStore
Profiles                      : 0
RemoteDynamicKeywordAddresses : {}
RuleGroup                     :
StatusCode                    : 65536
PSComputerName                :
CimClass                      : root/standardcimv2:MSFT_NetFirewallRule
CimInstanceProperties         : {Caption, Description, ElementName, InstanceID...}
CimSystemProperties           : Microsoft.Management.Infrastructure.CimSystemProperties
```

Figure 8.2f — Detailed rule properties confirming inbound SMB traffic is blocked on TCP 445.

Step 9 — Stealth Verification Scan (Kali)

Purpose

Perform a stealth-style TCP SYN scan with -Pn to confirm the host is no longer responding to typical discovery or probe traffic after hardening. This scan simulates what an external attacker would try when simple pings are blocked or filtered.

```
┌──(parallels㉿kali-linux-2025-2)-[~]
└─$ sudo nmap -Pn -sS -p 135,139,445,5040 -v 192.168.68.71 -oN ~/Labs/VulnScan/phase9_Pn_stealth_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-14 15:03 EDT
Initiating ARP Ping Scan at 15:03
Scanning 192.168.68.71 [1 port]
Completed ARP Ping Scan at 15:03, 1.43s elapsed (1 total hosts)
Nmap scan report for 192.168.68.71 [host down]
Read data files from: /usr/share/nmap
Nmap done: 1 IP address (0 hosts up) scanned in 1.49 seconds
           Raw packets sent: 2 (56B) | Rcvd: 0 (0B)
```

Figure 9 — Stealth Nmap scan from Kali showing the Windows host reported as "down" (probes blocked), confirming successful network-level hardening.

Outcome / Interpretation

The Nmap output reports the target **as host down** (Nmap: "1 IP address (0 hosts up)"), and the ARP ping shows no reply to probe packets. This indicates the host is either dropping probe packets (firewall blocking) or actively configured not to respond which was the expected result after the firewall rule and SMB hardening. In short: **external probes cannot reach SMB-related ports anymore**.

Conclusion & Lessons Learned

This lab demonstrated a complete vulnerability management workflow — from discovery and enumeration to remediation and validation. After applying SMB and firewall hardening, all previously reachable services were confirmed blocked from external access.

Key takeaways include:
- SMBv1 should remain permanently disabled across Windows systems.
- Firewall enforcement is the first line of defense against lateral movement.
- Post-hardening validation (rescan) is critical to confirm mitigation success.