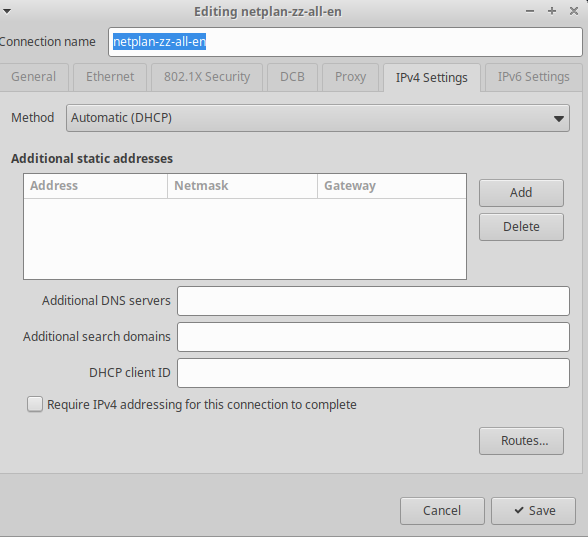
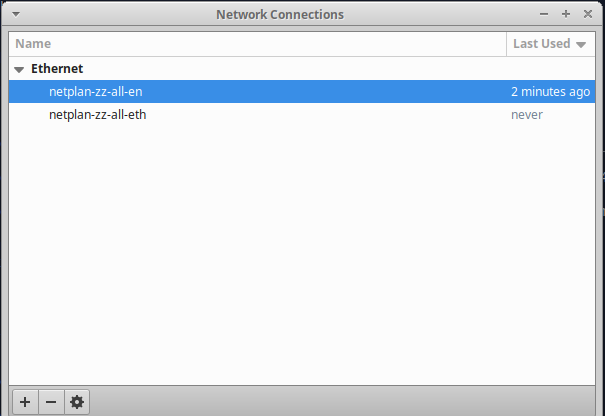
# Lab Setup Requirements

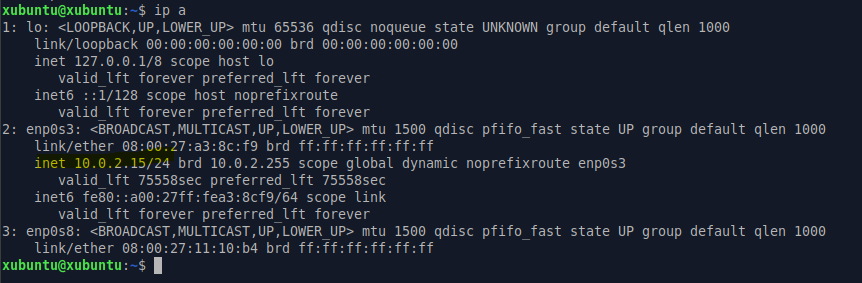
* **Ubuntu (attacker)** with tools like smbclient, nmap, enum4linux, crackmapexec, or impacket.
* **Windows VM (target)** with SMB enabled and a shared folder.

Window VM IP address: 10.0.2.15, adapter 2 internal network ip address is 192.168.50.11

Below is how to manually assign static ip to xubuntu vm



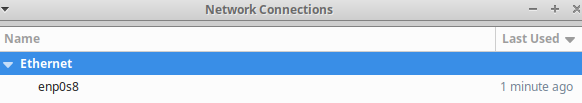


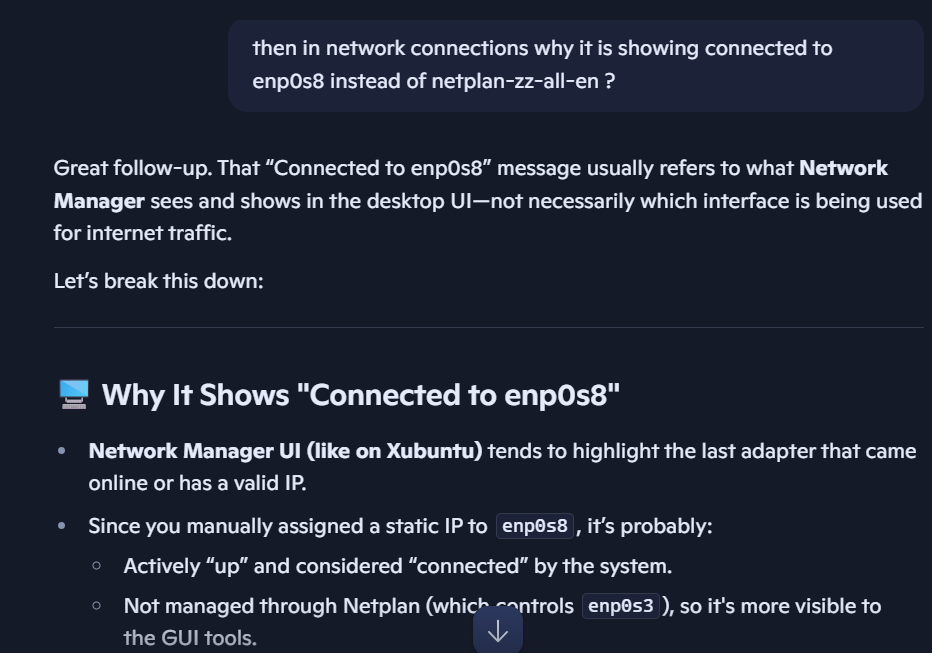


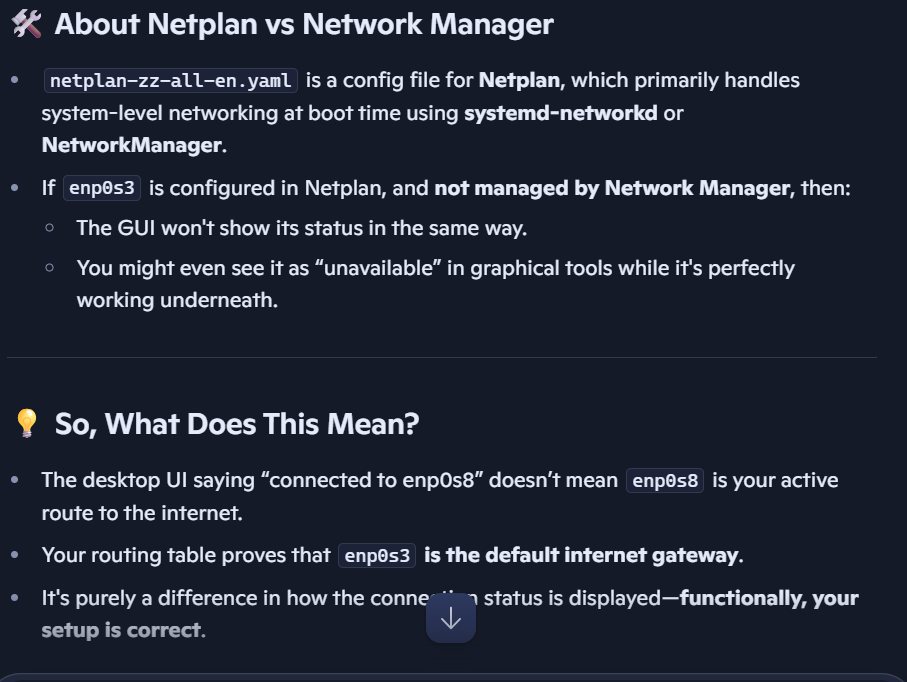
So, when I manually assigned static ip add to adapter2 (internal network enp0s8) on xubuntu as seen below

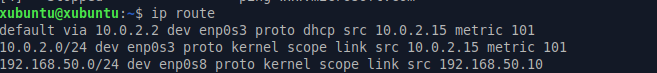


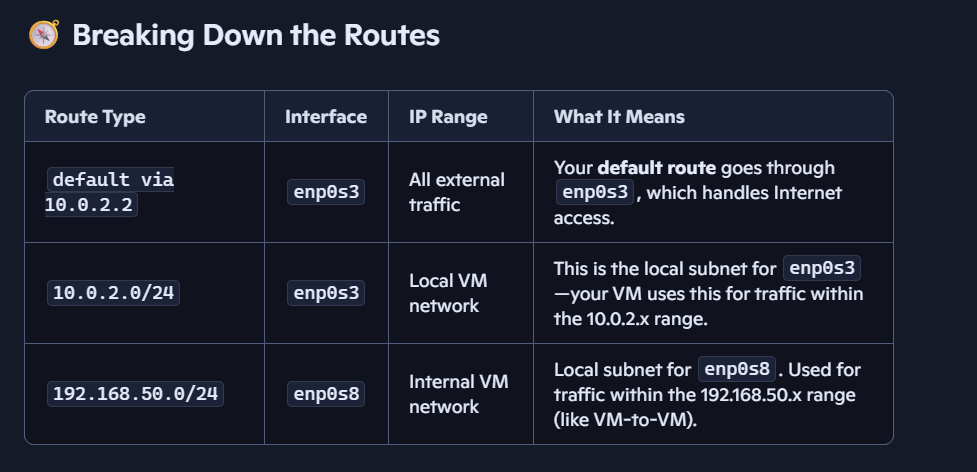
The network connection shows that the xubuntu connected to adapter 2

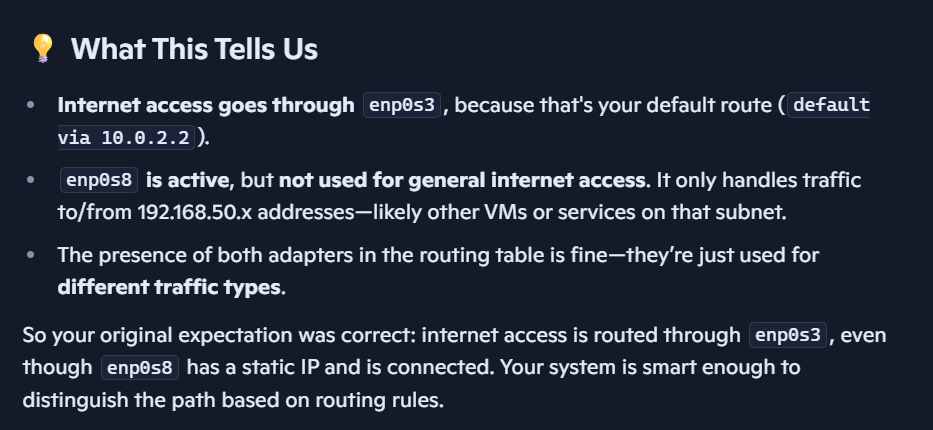


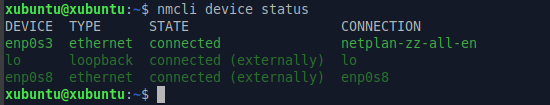


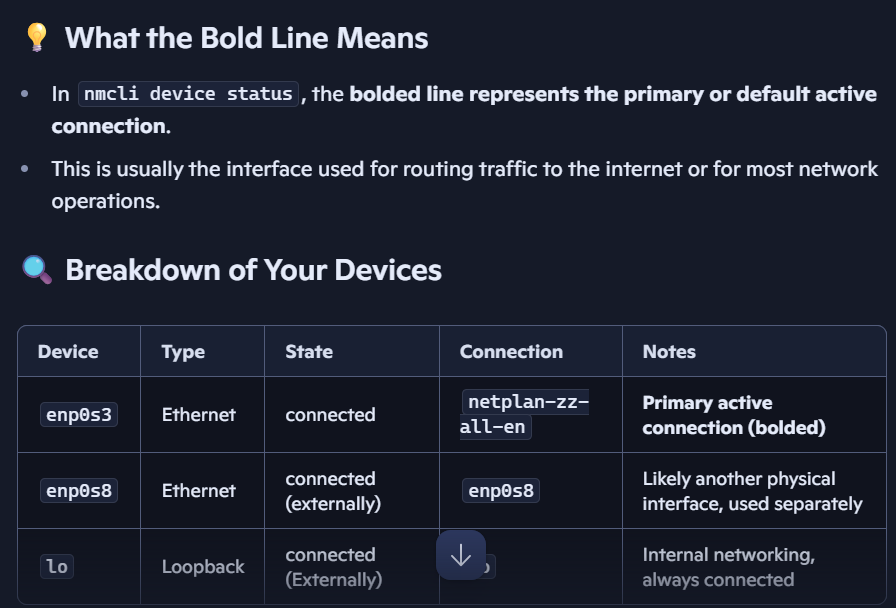


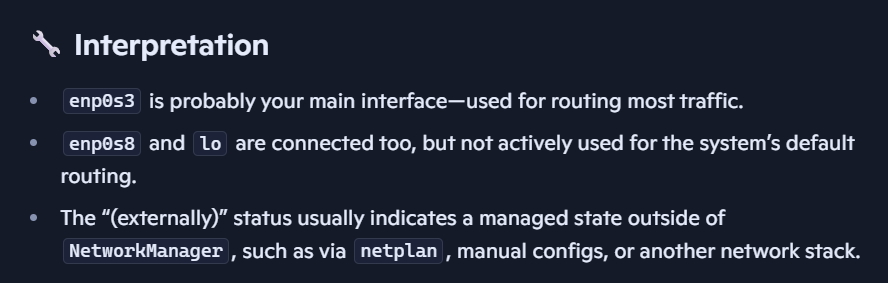


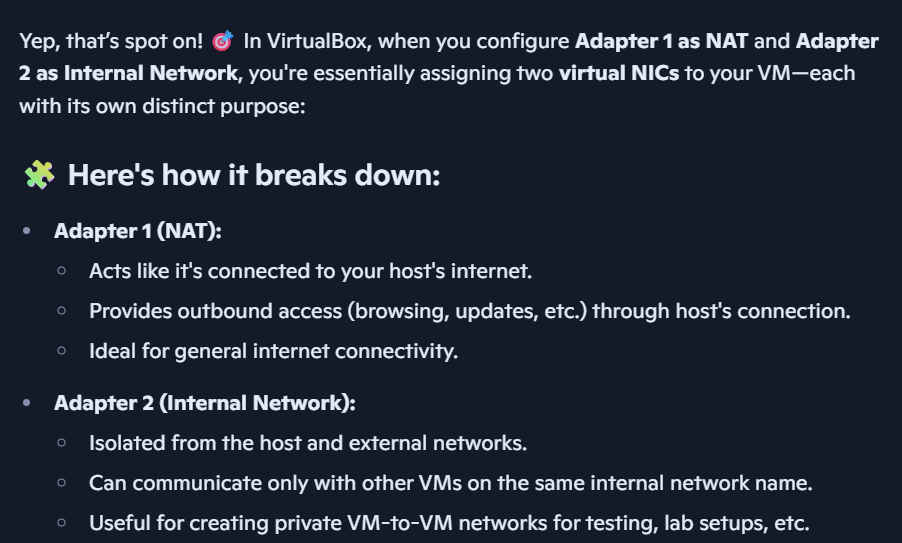












**Mandatory Checks on the Windows VM (The Server)**

The settings you were asked to check are all Windows-side configurations. They are crucial for a successful SMB connection, regardless of whether the client is a Linux or Windows machine.

* **Firewall (WF):** This is the most common culprit. You must ensure the Windows Firewall has rules that allow inbound traffic on TCP ports 139 and 445. The "Public to Private" change you mentioned is directly related to this. Windows Firewall applies different rules to Public and Private networks, with Public being much more restrictive. If your network adapter is set to Public, the firewall is likely blocking SMB traffic.
* **Services.msc:** For SMB to work, the "Server" service (and its dependencies) must be running. You can check this in the Services app.
* **SMB Protocol Version (regedit/gpedit):** The SMB protocol has evolved (SMB1, SMB2, SMB3). By default, modern Windows versions disable SMBv1 for security reasons. If your smbclient on Ubuntu is an older version that only supports SMBv1, you will need to either update your client or, for testing purposes, enable SMBv1 on the Windows machine. This is where you would use gpedit.msc (Local Group Policy Editor) or regedit.exe.
* **Credential/Security Policies (secpol/gpedit):** These settings control authentication. If your user account is locked out, disabled, or if there are specific policies that prevent network logons, the connection will fail with a "logon failure" error. secpol.msc and gpedit.msc are the tools to check this.
* **File Sharing:** Is the folder you're trying to access actually shared, and does the user you're trying to authenticate with have the correct permissions (both share permissions and NTFS permissions)?

**Mandatory Checks on the Ubuntu VM (The Client)**

Your Ubuntu VM also has some key settings that need to be in place.

* **Samba Client Utilities:** You need to have the smbclient package installed.
  + sudo apt update
  + sudo apt install smbclient
* **The Command Syntax:** The command you use must be correct.
  + smbclient -L //<Windows\_VM\_IP\_Address> -U <username>
  + The -L flag lists shares, which is a great first test.
  + Make sure you use a valid username that exists on the Windows VM. The password will be prompted for after you run the command.

**In Summary: The Thumb Rule for Troubleshooting**

To test smbclient from Ubuntu to a Windows VM correctly, follow this step-by-step thumb rule:

1. **Network Reachability:**
   * Check that both VMs are on the same virtual network and their network adapters are configured to allow communication.
   * From Ubuntu, ping the Windows VM's IP address. If this fails, fix the networking/firewall first.
2. **Port Availability:**
   * From Ubuntu, use nmap or nc to verify that TCP ports 139 and 445 are open on the Windows VM. If not, disable the Windows Firewall for the private profile *for testing purposes*, then enable it and create an inbound rule for SMB.
3. **Windows-Side Configuration:**
   * On the Windows VM, confirm the shared folder is correctly configured with the necessary share and NTFS permissions.
   * Verify that the "Server" service is running.
4. **Client-Side Execution:**
   * Ensure smbclient is installed on the Ubuntu VM.
   * Run the smbclient command with the correct syntax, using a valid Windows user account. Start with smbclient -L to list shares.

By following these steps, you'll systematically eliminate potential issues from the network layer up to the application layer and isolate the root cause of the problem

# Step 1: Configure SMB on Windows (Target)

You’ll need a shared folder on the Windows VM to act as bait for access and exfiltration.

## Enable SMB

* Go to **Control Panel → Programs → Turn Windows features on or off**
* Ensure **"SMB 1.0/CIFS File Sharing Support"** is checked

**🔹 Create a Shared Folder**

1. Create a folder: C:\SharedData
2. Right-click → Properties → “Sharing” tab
3. Click “Advanced Sharing”
4. Enable “Share this folder”, name it SharedData
5. Set permissions:
   * Add Everyone with Read/Write
6. Optional: Create a dummy file inside: loot.txt

# Step 2: Install SMB Client Tools on Ubuntu

Before you can access SMB shares, you need the right tools installed.

Sudo apt update

Sudo apt install smbclient

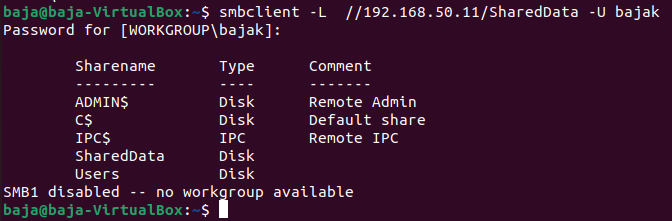
# Step 3: Discover the SMB share

Use smbclient to list available shares on the Windows VM.

smbclient -L //<WINDOWS\_VM\_IP> -U <username>

you will be prompted for the password as seen in the screen shot below

if successful, you’ll see a list of available shares including the SharedData folder



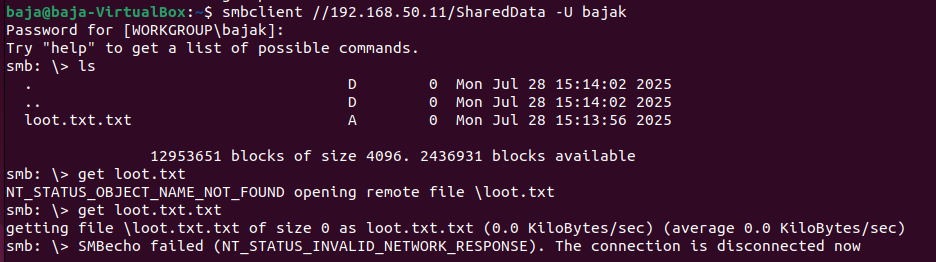
# Step 4: Access the share using

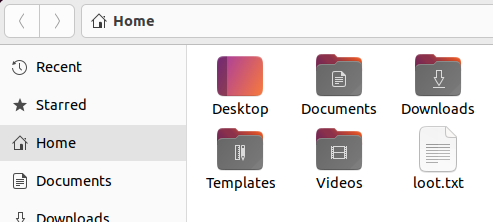
You can interact with the share directly using the command line:

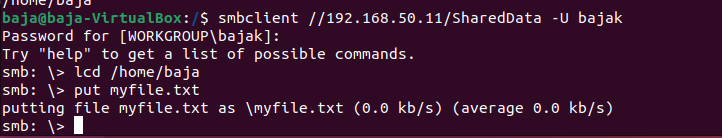
smbclient //<WINDOWS\_VM\_IP>/SharedData -U <username>

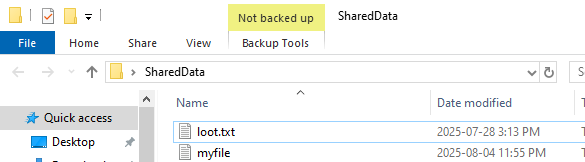
Once inside the interactive shell, you can run commands like:

* ls — list files
* get loot.txt — download a file
* put myfile.txt — upload a file
* exit — quit









Events 5140 and 5145 are generated by the "Audit File Share" and "Audit Detailed File Share" policies. These policies monitor connections to the share itself.

However, Event ID 4663 is generated by the **"Audit File System"** policy, and it requires both the policy to be enabled and the specific file or folder to be configured for auditing

