Jelly Concepcion

Aspiring Cyber Defense Incident Responder | Future Chief Information Security Officer (CISO) | Passionate About Cybersecurity & Threat Detection

Email: jellydizonconcepcion@gmail.com | GitHub: https://github.com/jellyconcepcion/blue-team-homelab | LinkedIn: https://www.linkedin.com/in/jellyconcepcion/

Introduction:

> This guide provides a detailed, step-by-step process to set up the Windows 11 VM in my Blue Team lab environment (Month 1). It includes installation, network configuration, Wazuh agent deployment, verification, and evidence capture for logging and monitoring.
>
> **Goal:** Create a Windows endpoint ready to communicate with the Ubuntu SIEM server, enabling real-world log collection and SOC lab exercises.
>
> **Audience:** SOC analysts, cybersecurity students, and enthusiasts looking to replicate a Windows endpoint integrated with a SIEM environment.

Create a Windows VM using Win11_24H2_English_x64.iso.

- Name and Operating System
  Name: WIN11-LAB-01
  ISO Image: Win11_24H2_English_x64 (C:\Users\Concepcion\Documents\blue-team-homelab\iso\Win11_24H2_English_x64.iso)
  Edition: Windows 11 Pro (10.0.26100.1742 / x64 / en-US)
  Type: Microsoft Windows
  Version: Windows 11 (64-bit)
  Screenshot: blue-team-homelab/01-lab-setup/02-windows-endpoint/screenshots/windows-create-vm-name-os.png

- Unattended Install
  Username: labuser
  Password: Dream!110100
  Hostname: WIN11-LAB-01

Screenshot: windows-create-vm-unattended-install.png

- Hardware (Note: 2 CPU cores is a must to install the Win11_24H2_English_x64 Windows 11 Pro)
  Base Memory: **4096** MB
  Processors: 2 CPU cores (Win11_24H2_English_x64 Windows 11 Pro need at least 2 core to be compatible with the device)
  [Checked] Enable EFI (special OSes only)
  Screenshot: windows-create-vm-hardware.png

- Hard Disk
  [Selected] Create a Virtual Hard Disk Now
  Hard Disk File Location and Size: **60** GB
  Hard Disk File Type and Variant: VDI (Virtual Disk Image)
  [Unchecked] Pre-allocate Full Size
  Screenshot: windows-create-vm-hard-disk.png

- Windows VM → Settings → System → Motherboard
  Boot Order:
  Optical (checked, top priority)
  Hard Disk (checked, second)
  Floppy (unchecked)
  Network (unchecked)
  TPM: v2.0
  [Checked] Enable EFI (special OSes only)
  Screenshot: windows-vm-settings-system.png

- Windows VM → Settings → Network
  Adapter 1: [Checked] Enable Network Adapter = NAT (Screenshot: windows-vm-settings-adapter1.png)
  Adapter 2: [Checked] Enable Network Adapter = Host-only Adapter (Screenshot: windows-vm-settings-adapter2.png)

- Windows VM → Settings → Storage
  Controller: SATA
    - WIN11-LAB-01.vdi
    - Win11_24H2_English_x64.iso
    - Unattended…

  Remove the “Unattended…” by double clicking it → **Remove Attachment**
  Screenshot: windows-vm-settings-storage.png

Before starting the Windows VM, screenshot the summary of Windows VM on vbox VMs list: windows-vm-overall-setup-summary.png

---

- Start the windows vm
- Once you see “Press any key to boot from CD or DVD……”, you need to press any key immediately so that it will proceed to the windows setup.

Windows Setup

- Select language settings: English (Philippines)
- Select keyboard settings: US
- Select setup options: Choose “Install Windows 11”. Check “I agree…”
- Product key: Click “I don’t have a product key”
- Select Image: Choose “Windows 11 Pro”
- Applicable notices and license term: Click “Accept”
- Select location to install Windows 11: Just click “Next”. Screenshot: win11-storage-selection.png
- Ready to install: Click "Install". Screenshot: win11-install.png
- Installing Windows 11: “Your computer may restart a few times”
- Is this the right country or region?: Choose “Philippines”
- Is this the right keyboard layout or input method?: Choose “US”
- Want to add a second keyboard?: Click “Skip”
- Let’s name your device: Enter: WIN11-LAB-01 (Screenshot: win11-oobe-device-name.png)
- How would you like to set up this device?: Choose “Set up for personal use”. Screenshot: win11-oobe-setup-type.png
- Your update is in progress. We’ll take it from here.
- When at “Unlock your Microsoft experience”
    - turn off the win11 vm first
    - Go to Win11 VM → Settings → Network
    - Disable “Adapter 1: Enable Network Adapter = NAT”
    - Start the Win11 VM
- Is this the right country or region?: Choose “Philippines”
- Is this the right keyboard layout or input method?: Choose “US”
- Want to add a second keyboard?: Click “Skip”
- Let’s connect you to a network: Click “I don’t have internet”. Screenshot: win11-oobe-no-internet.png
- Who’s going to use this device?: Enter your name: labuser (Screenshot: win11-oobe-create-user.png)
- Create a super memorable password. Enter a password: Dream!110100 (Screenshot: win11-oobe-create-password.png)
- Now add security questions: Just fill out. Screenshot: win11-oobe-security-question.png
- Choose privacy settings for your device: Keep all defaults (ON) and just click “Accept”. Screenshot: win11-oobe-privacy-settings.png
- Once inside the win11 vm:

- turn off the win11 vm first
- Go to Win11 VM → Settings → Network
- Enable “Adapter 1: Enable Network Adapter = NAT”
- Start the Win11 VM

- Within the created win11 vm, navigate to the upper left → **Devices** → **Install Guess Additions CD image**
  - **Go to File Explorer**
  - Click “**CD Drive (D): VirtualBox Guess Addtions”. Screenshot: Screenshot the File Explorer showing `CD Drive (D:) VirtualBox Guest Additions` mounted. Filename: `win11-guest-additions-cd.png`**
  - **Double click “VBoxWindowsAdditions” application. Screenshot: Screenshot the VBoxWindowsAdditions Setup wizard window open (any step where it clearly says “VirtualBox Guest Additions”). Filename: `win11-guest-additions-setup.png`**
  - **Just click “Next” then “Install”**
  - **Select “Reboot Now” then click “Finish”. Screenshot: Screenshot the final screen that says “Reboot Now” with the checkbox and “Finish” button. Filename: `win11-guest-additions-reboot.png`**
  - **Wait until it totally reboot.**

Steps to Verify Guest Additions Installed

1. **Log in to your Windows 11 VM (`labuser / Dream!110100`).**
2. **Right-click the Start menu (bottom left Windows icon).**
3. **Select Device Manager.**
4. **In Device Manager, expand Display adapters.**
5. **You should see:**

   VirtualBox Graphics Adapter (WDDM)

Screenshot: Have Device Manager open with Display adapters expanded so `VirtualBox Graphics Adapter (WDDM)` is clearly visible. Save it as: `win11-guest-additions-verified.png`

Check which adapter is which

Run in PowerShell (Admin):

Get-NetIPAddress -AddressFamily IPv4 | Format-Table InterfaceAlias,IPAddress

The NAT adapter will have something like `10.x.x.x or 192.168.1.x` with a **Default Gateway**.

The Host-Only adapter should be `169.254.x.x` or blank (before you assign static).

What to look for (based on what you reported):

- `Ethernet 2` should show `10.x.x.x or 192.168.1.x` → this is the **NAT** adapter.
- `Ethernet` should show `169.254.x.x` → this is the **Host-Only** (link-local) adapter.

Assign the correct static Host-Only IP to Host-Only adapter:

Now add the static `192.168.56.20` to the Host-Only adapter. **Do not** set a default gateway.

Run:

New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress 192.168.56.20 -PrefixLength 24

Set DNS for that adapter (optional, okay to add)

Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses 8.8.8.8

What to expect:

- `Ethernet` will now show `192.168.56.20/24`. Default gateway field should be blank / none.

Final check (single-list summary commands)

After the steps above, run:

```
ipconfig /all
```

(show both adapters with their final addresses: `Ethernet 2` → 10.x.x.x + gateway, `Ethernet` → 192.168.56.20 and gateway blank)

Screenshot: `win11-ipconfig.png`

Basic connectivity tests (do these before installing the agent)

Do these FROM **WIN11-LAB-01** (PowerShell or cmd):

1. Ping the Wazuh manager (Ubuntu) over Host-Only:

```
ping 192.168.56.10 -n 4
```

- **Screenshot**: `win11-ping-wazuh.png` — show successful replies.

Do this before the ping connectivity test on ubuntu vm.

On WIN11-LAB-01: Add an inbound firewall rule for ICMP

Run in **PowerShell (Admin)** on `WIN11-LAB-01`:

New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4 -IcmpType 8 -Direction Inbound -Action Allow

This creates a permanent inbound ping rule.

Screenshot: win11-firewall-allow-icmp.png

After allowing ICMP on win11 vm, do these FROM **wazuh-lab-01 (Ubuntu)** to confirm two-way reachability:
ping -c 3 192.168.56.20 # ping WIN11 from Ubuntu
Screenshot: `ubuntu-ping-win11.png`

If connectivity is OK → proceed to install the Wazuh agent

Do this in the sequence below.

A) On **your host browser** (or in WIN11 if it has internet)

1. open Wazuh dashboard
2. **Deploy new agent**
3. **Choose "Windows MSI 32/64 bits"**
4. In the **Server address** field, enter the IP of your Wazuh server (your Ubuntu SIEM Host-Only adapter): 192.168.56.10 (Note: This is how the Windows agent will know where to send logs. Do **not** use localhost or your NAT IP — only the **host-only IP**.)
5. Agent Name: Leave blank (default = hostname `WIN11-LAB-01`).
6. Groups: Leave as **Default** for now (no need to select another group).
7. On "**Run the following commands to download and install the agent:" copy the command.**
8. **Open PowerShell as Administrator inside WIN11-LAB-01.**
9. **Run the generated command: `Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.12.0-1.msi -OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent /q WAZUH_MANAGER='192.168.56.10'`**
10. **Start the service: Still in PowerShell, run: `NET START WazuhSvc`**
11. **Verify service is running: Run: `Get-Service WazuhSvc`**
12. **Screenshot: win11-wazuh-agent-install.png**

Verify agent appears in Wazuh dashboard (host/browser)
On **your host/browser** (go to https://192.168.56.10 → Agents):

- Wait ≈ 30–60 seconds after service starts; agent should register automatically.
- **Screenshot**: `wazuh-agents-win11-active.png` — Agents list showing `WIN11-LAB-01` Active/Connected.
- **Screenshot**: `wazuh-agent-details.png` — Agent detail view (mask/blur registration key if shown).

- On the win11 vm lockscreen, attempt to enter 5 wrong passwords. Screenshot: What to capture: the Windows lock screen or login attempt (if you captured while testing). Not required — Wazuh evidence is stronger: win11-failedlogon.png
- After successful login on the wazuh, you'll be on the "overview".
- On "Overview", find the "agents summary" then click "Active".
- On the "Agents" list, click the name "WIN11-LAB-01". Screenshot: wazuh-agents-win11-active.png
- Once you're on "WIN11-LAB-01", click the "Threat Hunting" on the upper left. Screenshot: wazuh-agent-details.png
- On "Threat Hunting", you'll see the filter "manager.name: wazuh-lab-01" and "agent.id: 001". Screenshot: What to capture: Threat Hunting Dashboard (the summary view) showing Authentication failure = 5 and the applied filters (manager.name & agent.id visible): wazuh-threat-authfail-summary.png
- To add more to the filter, click the words "- Authentication failure -"
- After clicking the words "- Authentication failure -", click the 2 donut chart where the words "Logon Failure - Unknown user or bad password" and "authentication_failed" are. The words "rule.description: Logon Failure - Unknown user or bad password" and "rule.groups: authentication_failed" will be added to the filter.
- After clicking all that, click the "Events" tab on the upper left. That's where you'll see the table for the 5 wrong password attempts you did including the timestamp, agents.name, rule.description, rule.level, and rule.id. And the authentication failure count. Screenshot: What to capture: Threat Hunting → Events table showing the 5 hits, with columns visible: timestamp, agent.name, rule.description, rule.level, rule.id: wazuh-events-list.png
- On the same Threat Hunting → Events view where you already see the 5 hits, click one of the event rows to open the "Document Details" and it will show a table tab and a json tab, as well as 2 clickable words " View surrounding documents" and "View single document". Screenshot: What to capture: Document Details → JSON for one representative event (the JSON you pasted). Make sure the part that shows "system.eventID": "4625", "targetUserName": "labuser", "logonType": "2" and "ipAddress": "127.0.0.1" is visible: wazuh-event-4625-json.png
- Go to the Document Details, click the "View single document" to expand the table view. Screenshots: What to capture: Document Details → Table view (if available) or expand the JSON into UI fields, showing rule.description, rule.id, firedtimes, data.win.eventdata.targetUserName, data.win.eventdata.logonType, etc.: wazuh-event-4625-table-1.png and wazuh-event-4625-table-2.png
- Go back to the Threat Hunting then Dashboard tab. On the search bar either type "data.win.system.eventID:"4625"" or "rule.id:"60122"" then click enter key or update button. Screenshot: What to capture: the Search/DQL bar with the exact query you used (e.g., data.win.system.eventID:"4625" or rule.id:"60122") and the results count/preview visible: wazuh-dql-search.png

---

Steps to Capture Event Viewer 4625 Failed Logon

1. **Log into your Windows 11 VM (`WIN11-LAB-01`).**
2. **Open Event Viewer**
    - Press **Win + R** → type `eventvwr.msc` → press **Enter**.
    - Or search in Start Menu: *Event Viewer*.
3. **Navigate to Security Logs**

In the left pane, expand:

```
Windows Logs → Security
```

    -
4. **Filter for Failed Logons (Event ID 4625)**
    - Right-click **Security** → choose **Filter Current Log...**

In the filter window, under **Event IDs**, enter:

```
4625
```

    -
    - Click **OK**.
    - Now you'll only see failed logon attempts.
5. **Find Your Events (same timestamp as Wazuh)**
    - Look for events at the exact times you attempted the wrong passwords.
    - Each event should have:
        - **Task Category**: Logon
        - **Keywords**: Audit Failure
        - **Event ID**: 4625

- **Account For Which Logon Failed** = your `labuser` account
- **Failure Information** = "Unknown user name or bad password"

Screenshot: What to capture: Windows Event Viewer → Security showing the 4625 entries locally on the VM (same timestamp)": win11-eventviewer-4625.png

---

Create the evidence file on Windows

(Note: You first need to cd "C:/Users/Concepcion/Documents/blue-team-homelab on gitbash then run git log --oneline and find the SHA you need.)

Run with PowerShell:

```
$sha = "eff3a26"

$timestamp = (Get-Date).ToUniversalTime().ToString("yyyy-MM-dd HH:mm UTC")

"blue-team-homelab commit: $sha | $timestamp" | Set-Content C:\Users\Public\lab-evidence-windows.txt

Get-Content C:\Users\Public\lab-evidence-windows.txt
```

Screenshot: windows-evidence-file.png

Take a VirtualBox snapshot (preserve state)

GUI method (VirtualBox)

1. Open VirtualBox, select the VM (e.g., `WIN11-LAB-01`).
2. Click **Snapshots** (right pane) → **Take** (camera icon).
3. Name: `M1-Win11-Evidence-eff3a26`
   Description: `Windows lab at commit eff3a26 (agent installed, evidence file added)`

Screenshot: vb-windows-snapshot.png