

Open in app ↗

 Search Write 7 F

Detection Engineering in a Homelab — Part 1: Setting Up Wazuh

 F

Frederick Adigun · 9 min read · Aug 20, 2025

 20

Detection engineering is all about creating, testing, and fine-tuning detection logic to spot malicious, suspicious, or policy-breaking activities across systems, networks, and applications.

In this homelab, I'll walk you through the setup of the tools that will form our detection stack:

- **SIEM:** Wazuh, an open source SIEM with built-in XDR capabilities
- **IAM:** Keycloak for identity and access management
- **Endpoint visibility:** Sysmon for deep system monitoring
- **Attack simulation:** Downloading and running a controlled malware sample on the endpoint, with detection rules configured to observe how the stack responds.

As a bonus project, I'll set up **pfSense** as a firewall and configure rules to manage traffic flow into and out of the network.

The write-up will be split into sections and uploaded all at once, each clearly labeled in sequence. You can check [my Medium page](#) for the next part after this one.

To keep things lively, each section will also include one or two completely unrelated facts that you absolutely do not need to know, but will give your brain a quick break from the technical deep dive. These will be marked with a pound sign (#) and written in italics.

As a prerequisite for this homelab, you should have VirtualBox installed. If you don't have it yet, you can quickly learn how to install it using this [link](#). You will also need a Windows VM in VirtualBox, where the Wazuh agent will be installed. You can learn to set one up [here](#).

In this first part, I will:

- Install the Wazuh server
- Access the Wazuh dashboard through a browser
- Install the Wazuh agent

Step 1: INSTALLING WAZUH SERVER

There are several ways to install the wazuh server, for this demonstration, I will install it through the Open Virtual Appliance (OVA) format. The OVA for Wazuh is a pre-built virtual appliance image that you can import directly into virtualization platforms VirtualBox, VMware or any other OVA compatible

virtualization systems to quickly spin up a Wazuh server without having to manually install and configure it.

Download the OVA file [here](#) (Fig. 1).

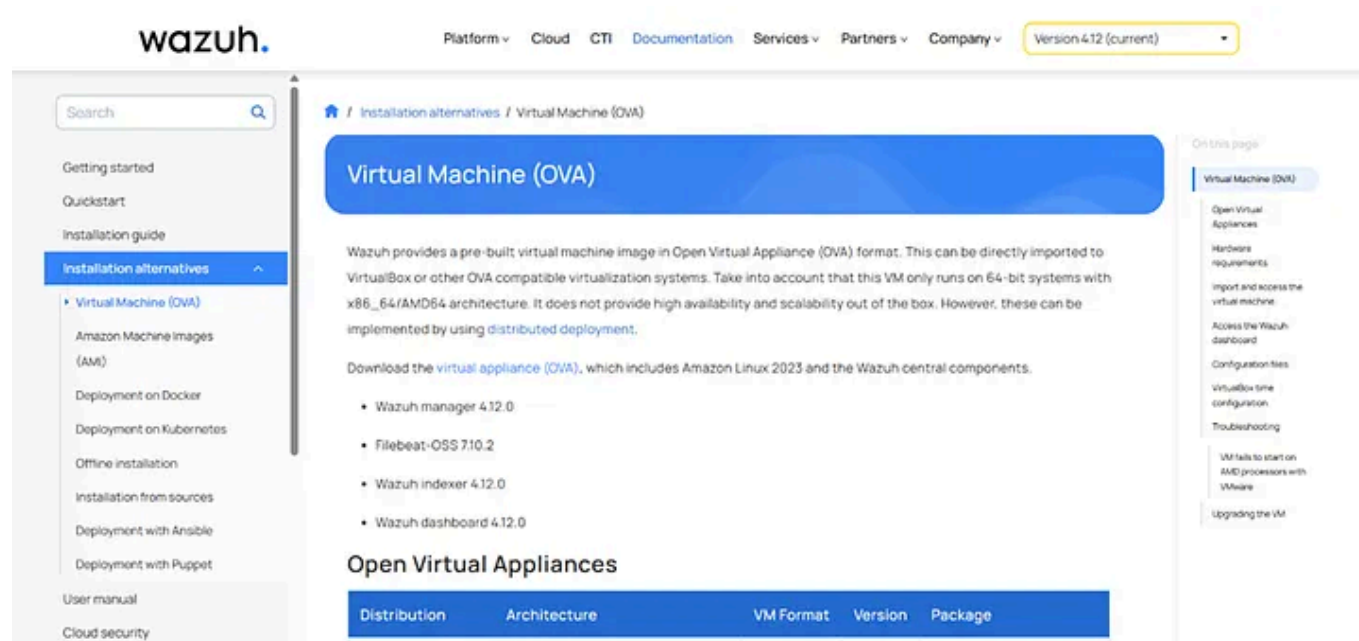


Fig. 1

After downloading the OVA file, right-click it, hover over **Open with**, and select **VirtualBox Manager** (Fig. 2).

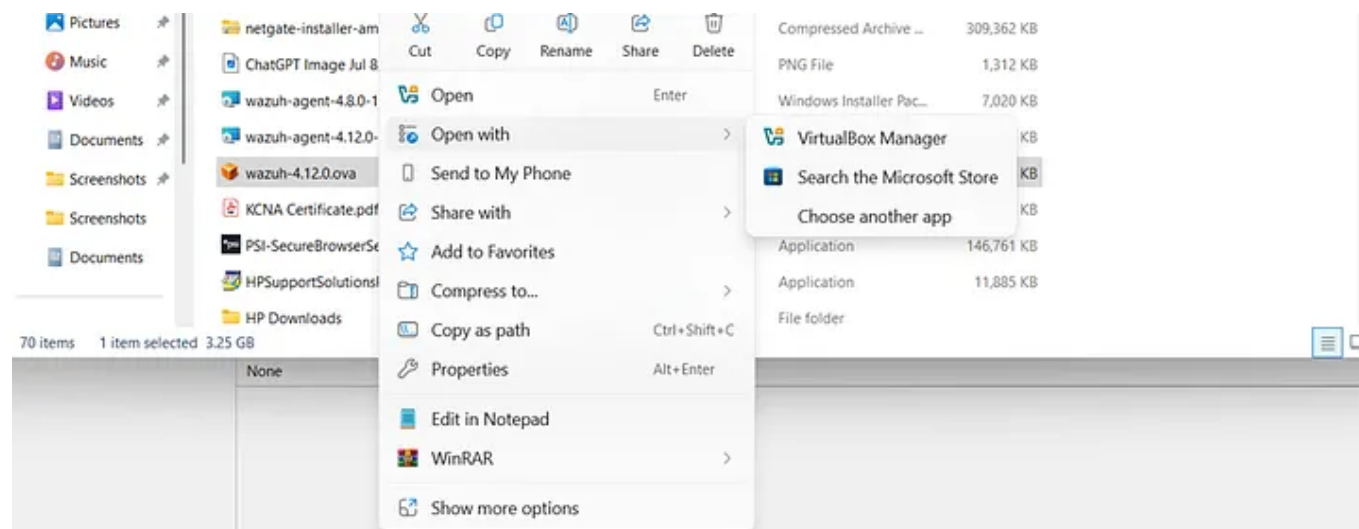


Fig. 2

On the VirtualBox Manager page that appears, click **Finish** to complete the import process (Fig. 3).

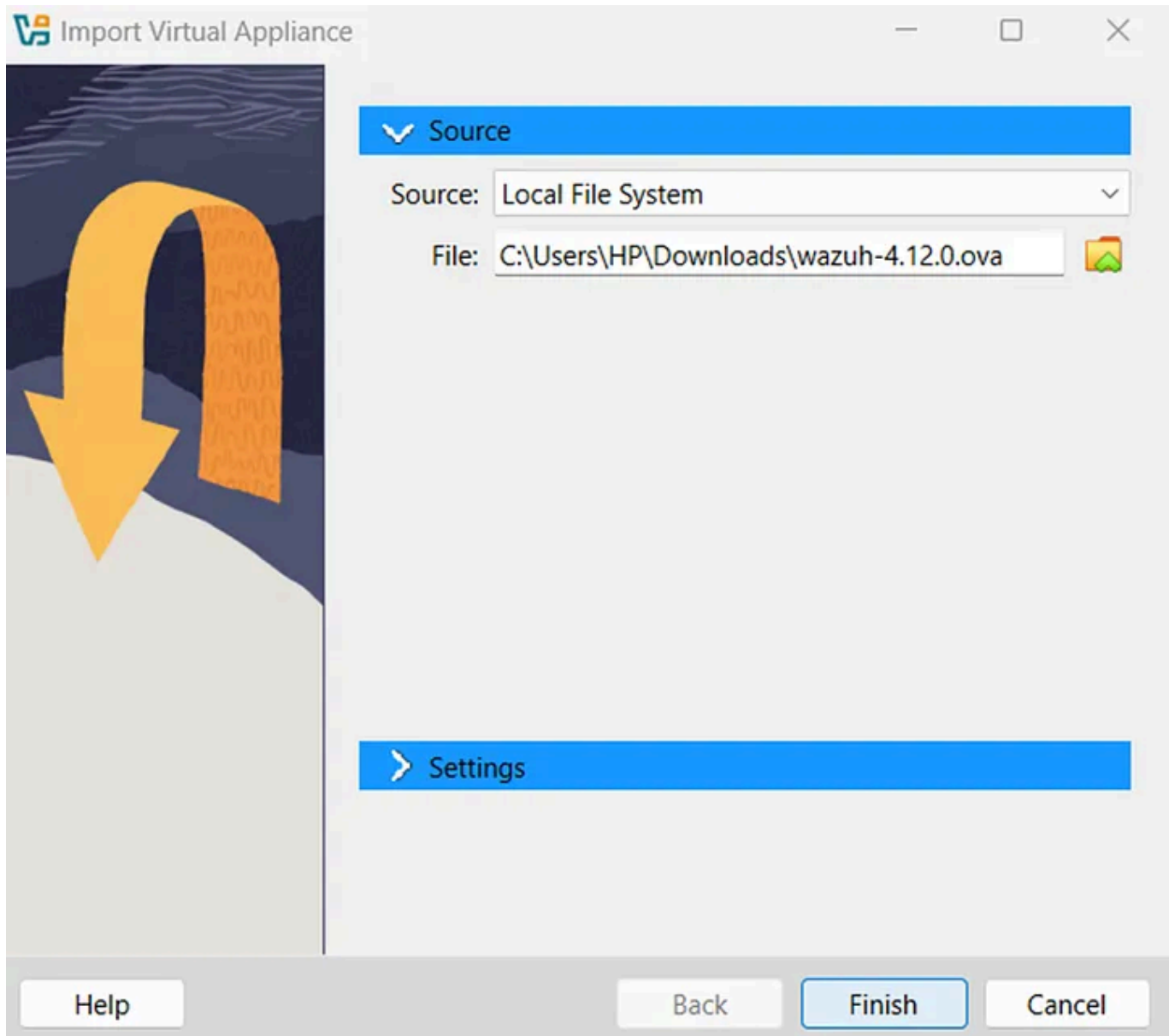


Fig. 3

Hold on for a few seconds while VirtualBox completes the process. You can monitor the progress in the upper-right corner of the screen (Fig. 4).



Fig. 4

After the import is complete, you'll need to make a few configuration changes. Wazuh recommends setting the graphic controller to **VMSVGA**; using another option can cause the VM window to freeze (Fig. 5).

To change this setting:

- Select the imported VM
- Click **Settings > Display**
- Under Graphics Controller, choose **VMSVGA**
- Click **OK** to save changes

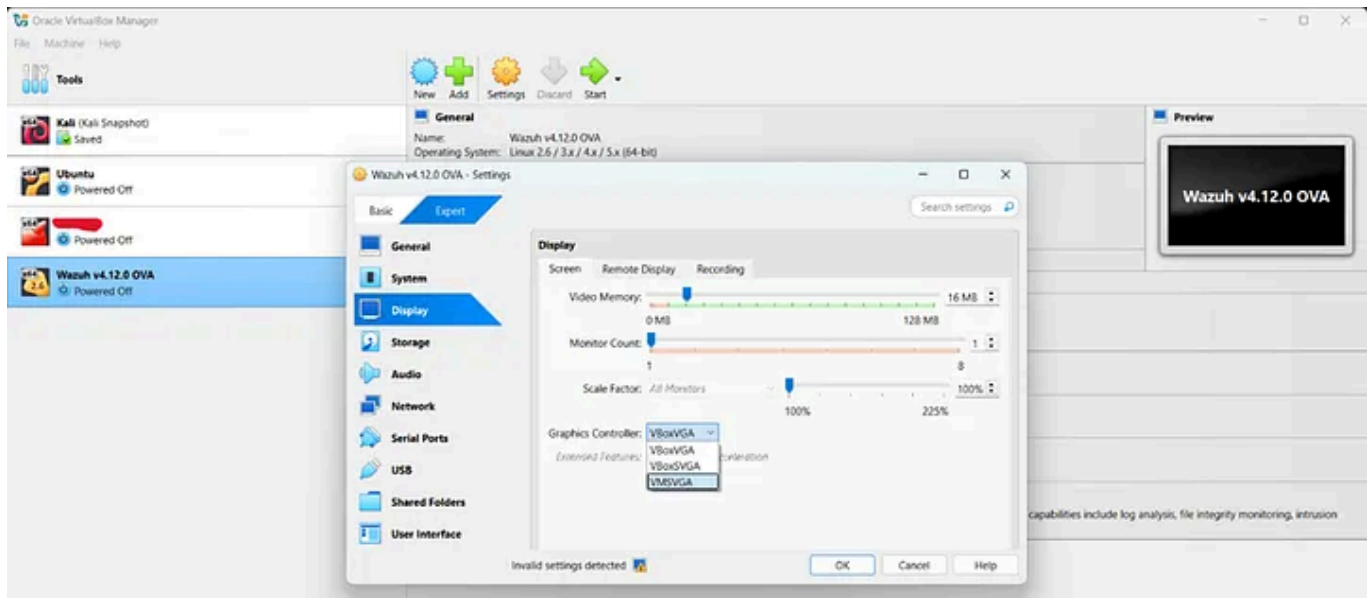


Fig. 5

After setting the graphics controller, adjust the resource allocation of the VM. Wazuh recommends 8 GB of RAM and 4 CPU cores, but using 4 GB of RAM and 2 CPU cores works fine too (Fig. 6).

To adjust the RAM:

- Click **Settings > System**
- Move the Base Memory slider to **4096 MB**

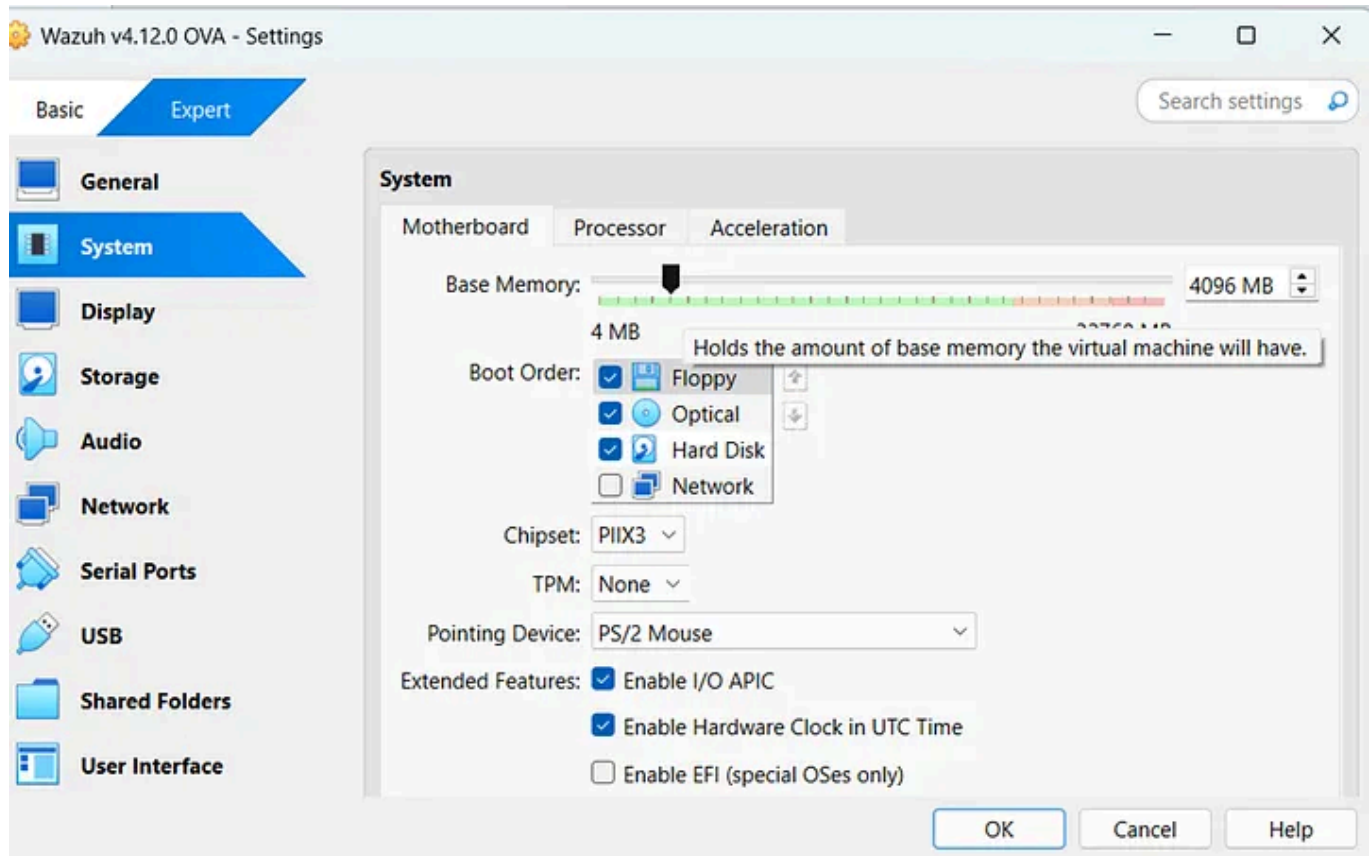


Fig. 6

Similarly, to adjust the number of CPU cores (Fig. 7):

- In the same **System** settings, switch to the **Processor** tab
- Move the slider to 2

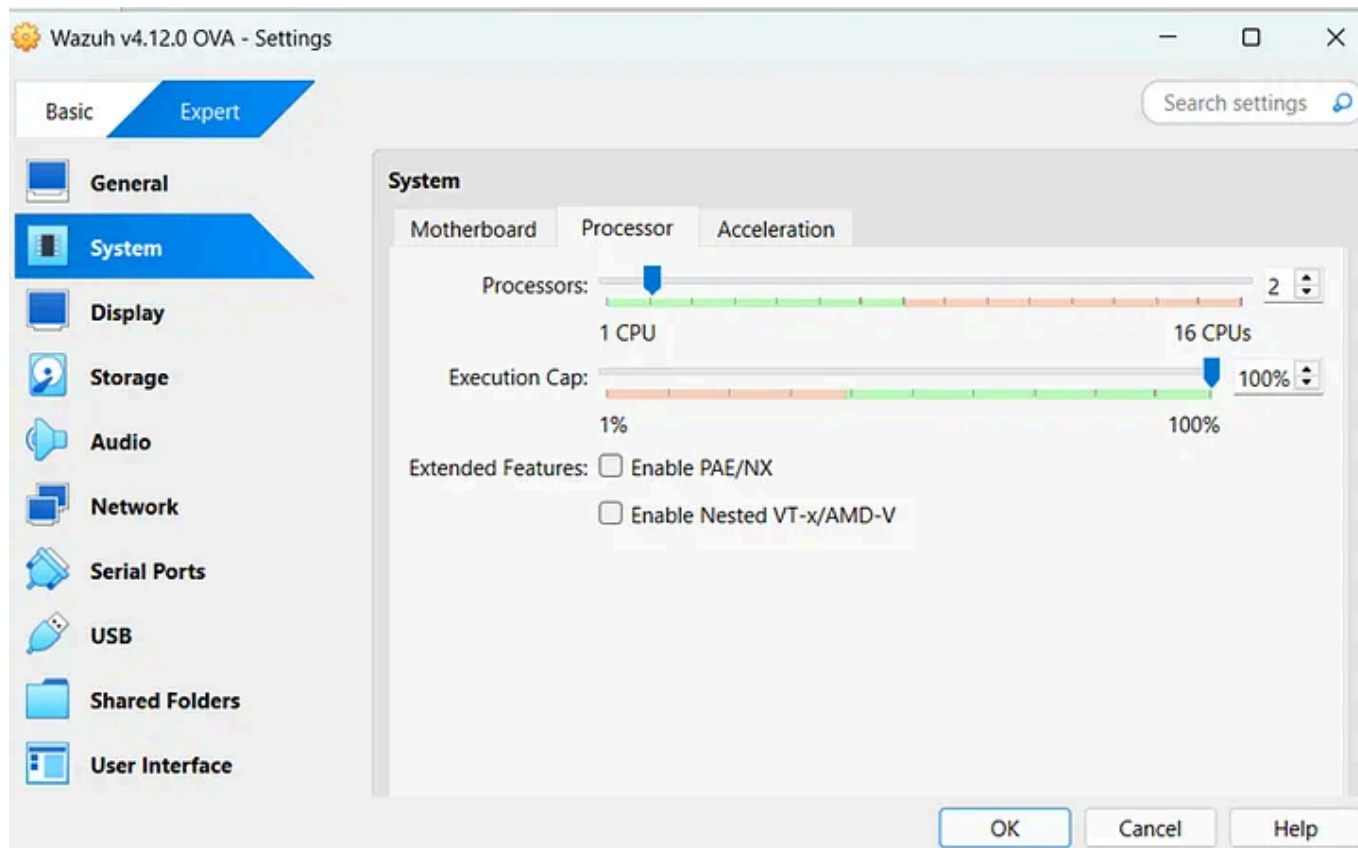


Fig. 7

Click OK to save changes.

Next, adjust the network settings (Fig. 8):

- Go to **Settings > Network**
- Change the **Attached to** option from **Bridged** to **NAT Network**

This isolates the VM from your physical network while still allowing internet access and communication with other virtual machines on the same NAT network. It provides a safer setup for testing and reduces exposure to external threats.

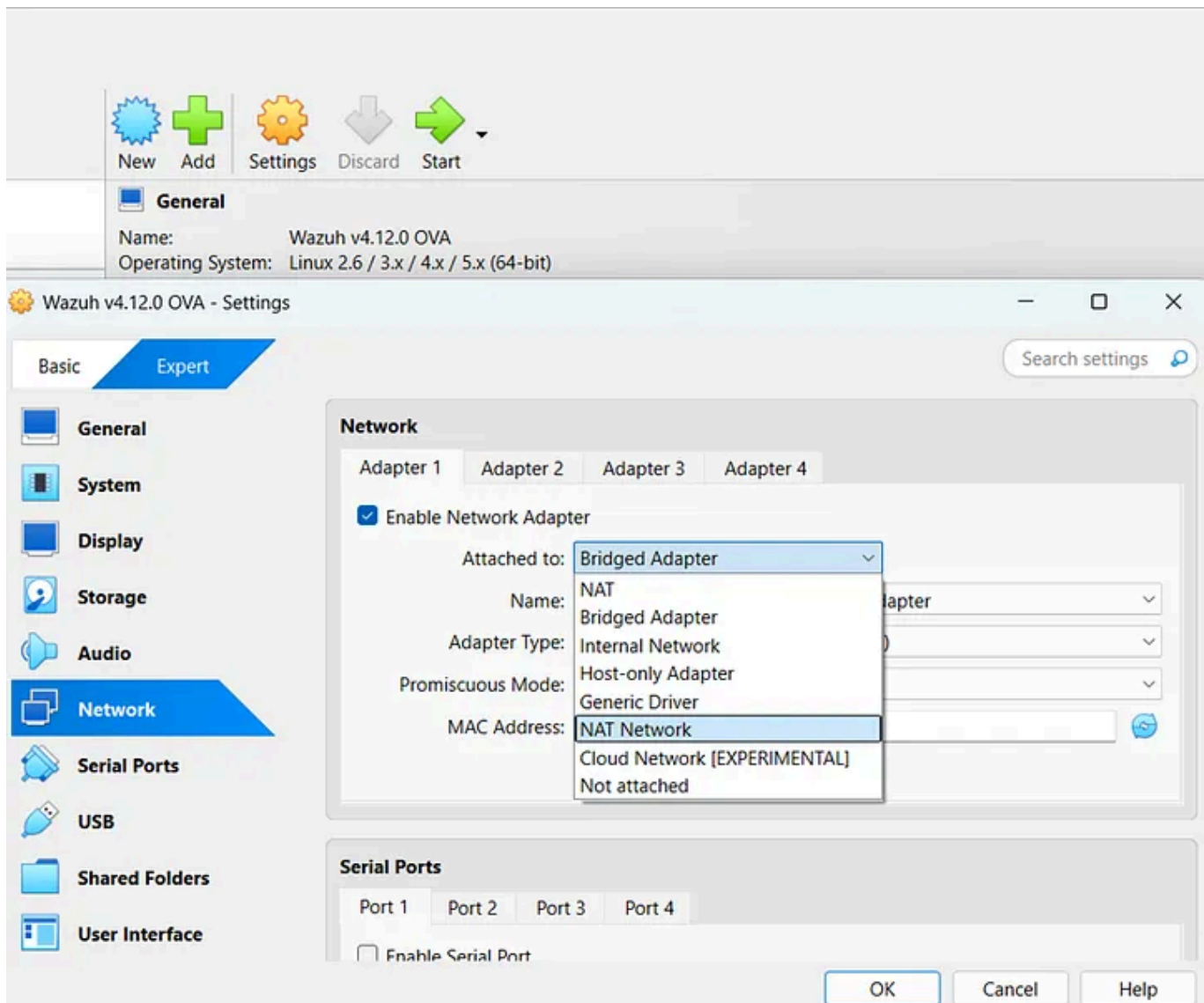


Fig. 8

Click OK to save the changes.

Starting the Wazuh VM

- Select the Wazuh VM
- Click **Start** (the green arrow pointing right) at the top of the screen

When the VM starts, you'll be prompted to log in. Use the default credentials (Fig. 9):

- **Username:** wazuh-user
- **Password:** wazuh

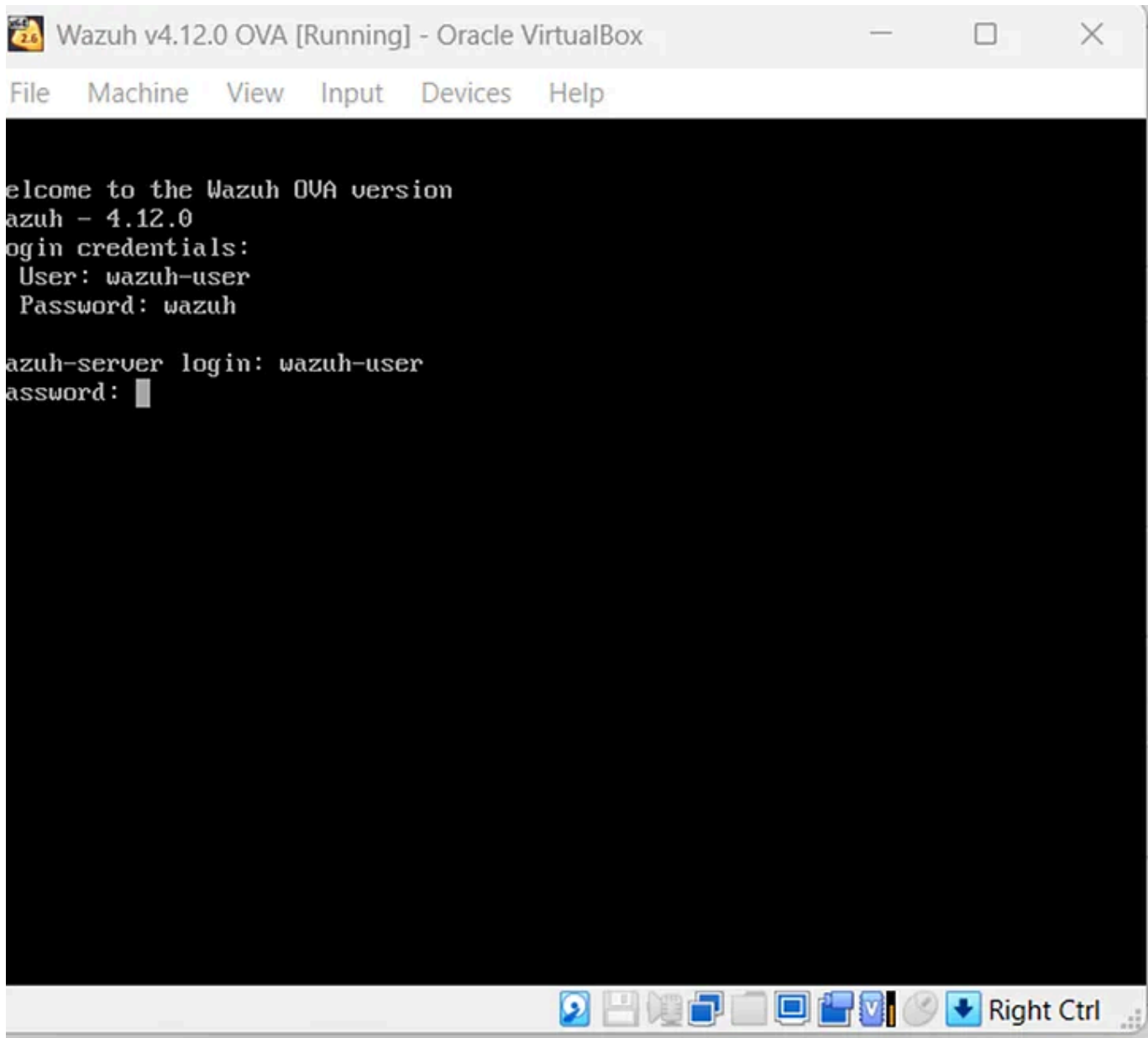


Fig. 9

Following successful login, this screen loads (Fig. 10):

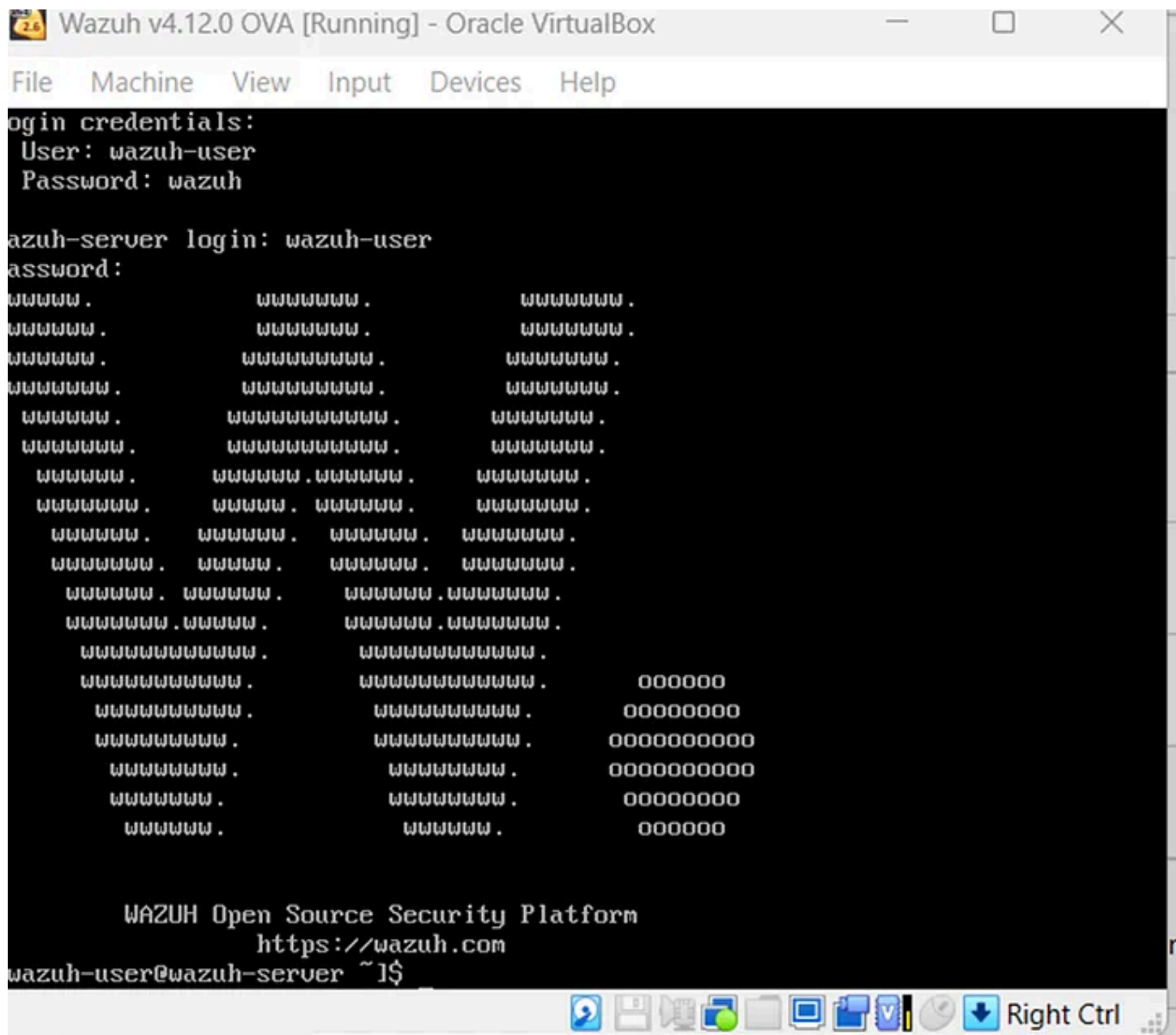


Fig. 10

The original “horsepower” measurement came from James Watt observing how much work a draft horse could do pulling beer kegs from a brewery.

Step 2: ACCESSING THE WAZUH DASHBOARD THROUGH A BROWSER

After logging into the Wazuh server, I can proceed to access the Wazuh dashboard. Next, run the `sudo -i` command (Fig. 11).

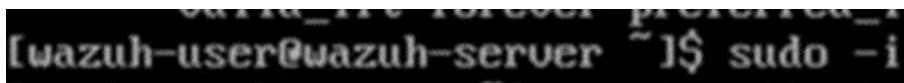


Fig. 11

This command is to run the subsequent commands as root.

After `sudo -i`, run the following commands (Fig. 12):

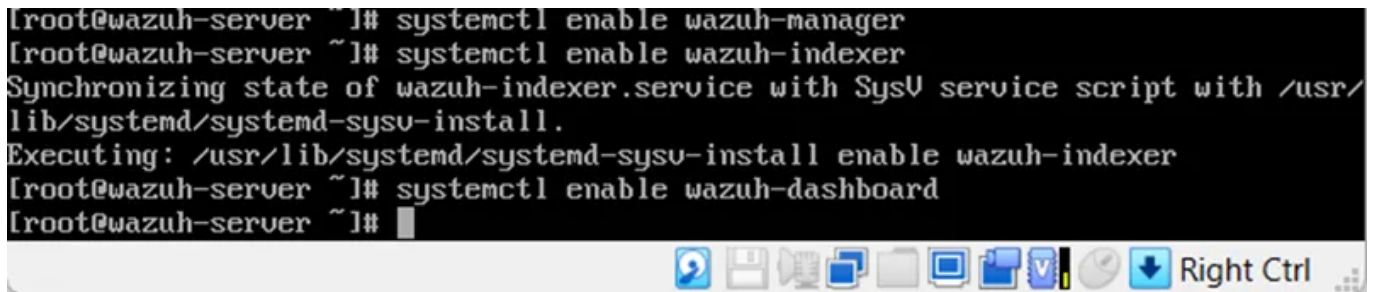
```
sudo systemctl enable wazuh-manager  
sudo systemctl enable wazuh-indexer  
sudo systemctl enable wazuh-dashboard
```

These commands ensure Wazuh's core services automatically start on boot, which is crucial for a fully functional setup.

- **`sudo systemctl enable wazuh-manager`** : Configures the Wazuh Manager, responsible for collecting, analyzing, and correlating security events, to launch at startup.
- **`sudo systemctl enable wazuh-indexer`** : Enables the Indexer, which stores and indexes logs so data can be searched and analyzed.
- **`sudo systemctl enable wazuh-dashboard`** : Sets the Dashboard, the web interface for visualizing alerts and managing the platform, to start automatically.

Running these commands right after installation is important because the dashboard relies on the manager and indexer to display real-time data. Without enabling them, the dashboard may not load correctly in the browser or show incomplete information. By enabling all services, you ensure that

after a reboot, Wazuh remains fully operational without manual intervention, providing continuous visibility and security monitoring.

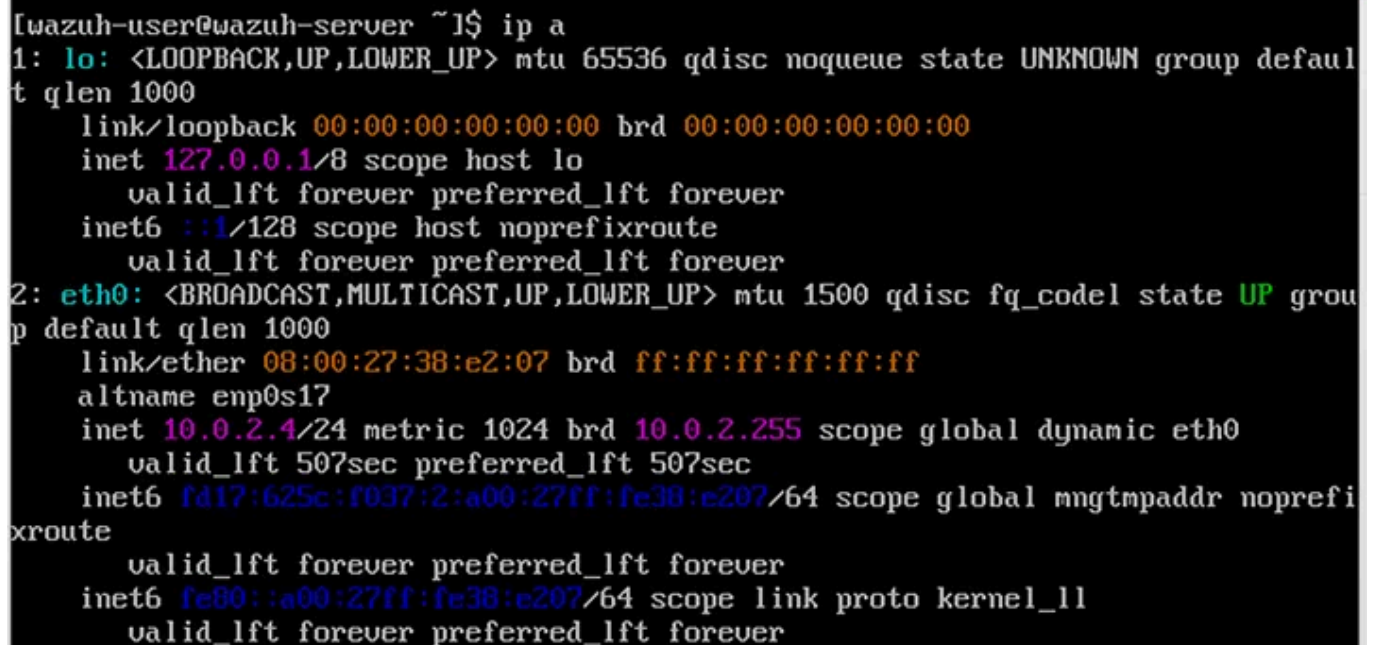
A terminal window with a black background and white text. The user is root@wazuh-server. The commands entered are: 'systemctl enable wazuh-manager', 'systemctl enable wazuh-indexer', and 'systemctl enable wazuh-dashboard'. The output for the first two commands shows the service state being synchronized with SysV and the systemd-sysv-install script being executed. The terminal window has a taskbar at the bottom with various icons and a 'Right Ctrl' button.

```
[root@wazuh-server ~]# systemctl enable wazuh-manager
[root@wazuh-server ~]# systemctl enable wazuh-indexer
Synchronizing state of wazuh-indexer.service with SysV service script with /usr/
lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable wazuh-indexer
[root@wazuh-server ~]# systemctl enable wazuh-dashboard
[root@wazuh-server ~]#
```

Fig. 12

Next, run the `ip a` command (Fig. 13)

The command `ip a` (short for `ip address`) is used to **display the IP addresses and network interface information** of your system.

A terminal window with a black background and white text. The user is wazuh-user@wazuh-server. The command entered is 'ip a'. The output shows details for the loopback interface 'lo' and the ethernet interface 'eth0'. The 'lo' interface has IP address 127.0.0.1. The 'eth0' interface has IP address 10.0.2.4. The terminal window has a taskbar at the bottom with various icons and a 'Right Ctrl' button.

```
[wazuh-user@wazuh-server ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:38:e2:07 brd ff:ff:ff:ff:ff:ff
    altname enp0s17
    inet 10.0.2.4/24 metric 1024 brd 10.0.2.255 scope global dynamic eth0
        valid_lft 507sec preferred_lft 507sec
    inet6 fd17:625c:f037:2:a00:27ff:fe38:e207/64 scope global mngtmpaddr noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe38:e207/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
```

Fig. 13

On another VM, open a browser and enter the IP address seen associated with inet (In my case 10.0.2.4).

You may see an SSL warning — click advanced and continue (Fig. 14).

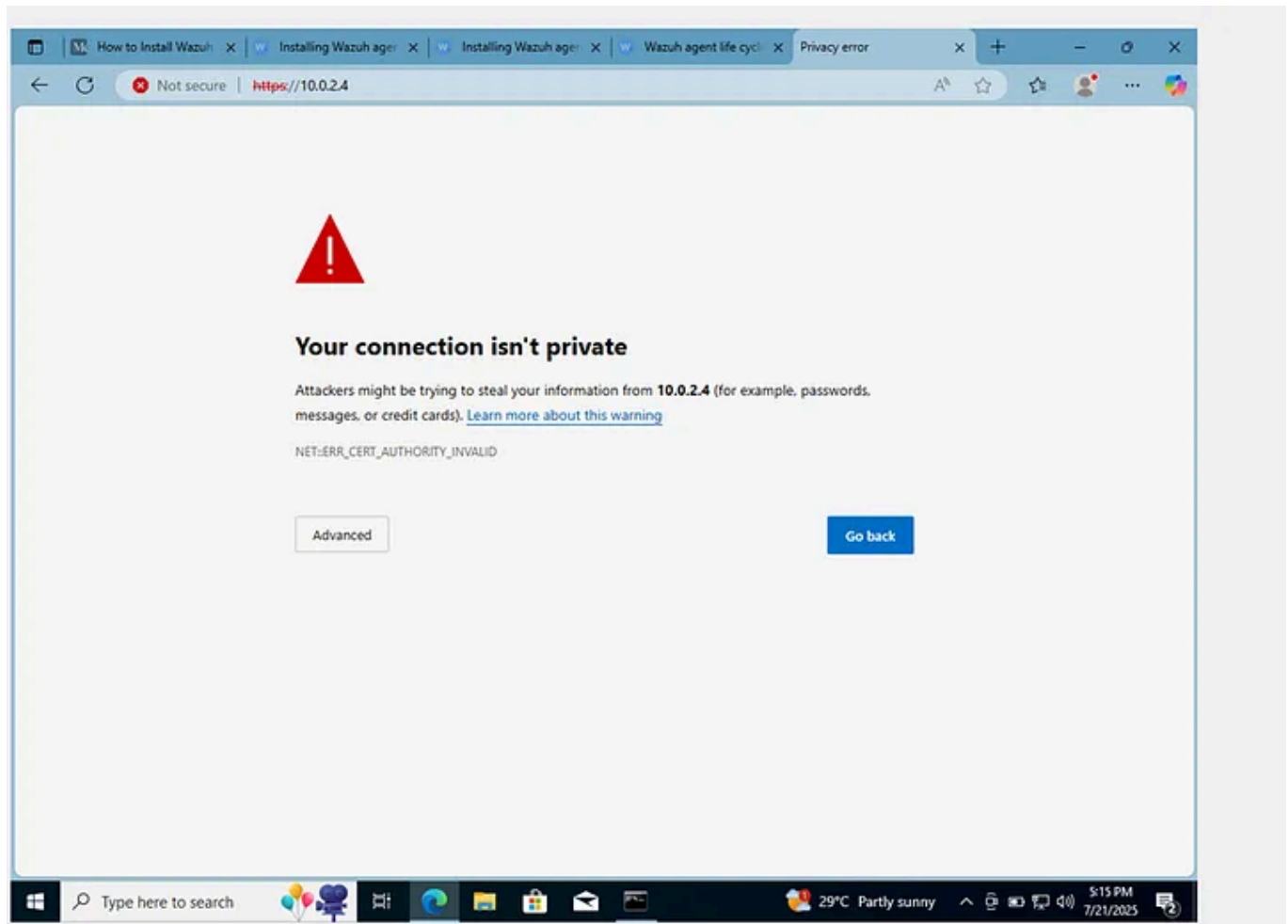


Fig. 14

You will then be presented with a login page, enter the following credentials (Fig. 15):

Username: admin

Password: admin

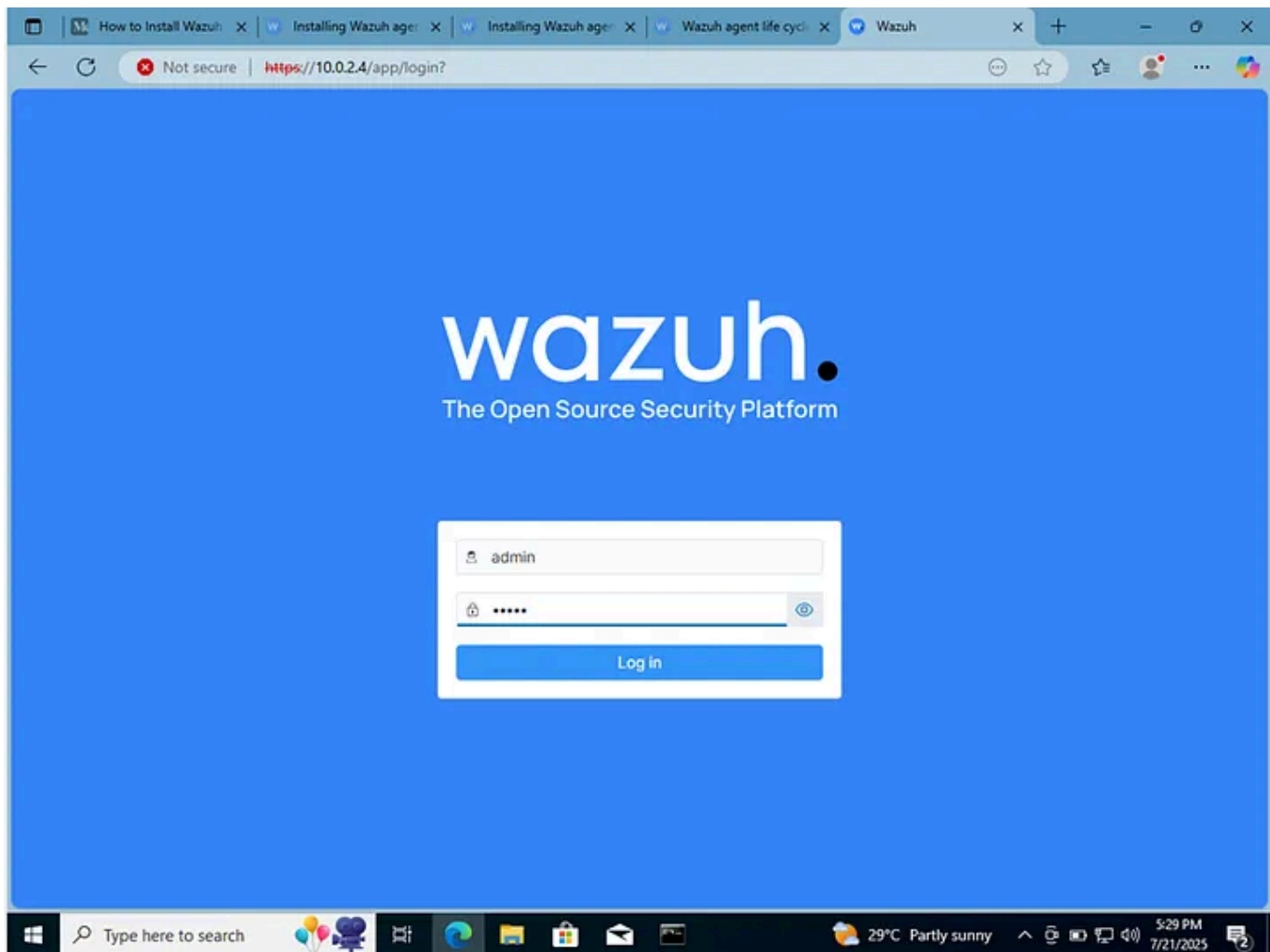


Fig. 15

After login, the wazuh dashboard displays (Fig. 16).

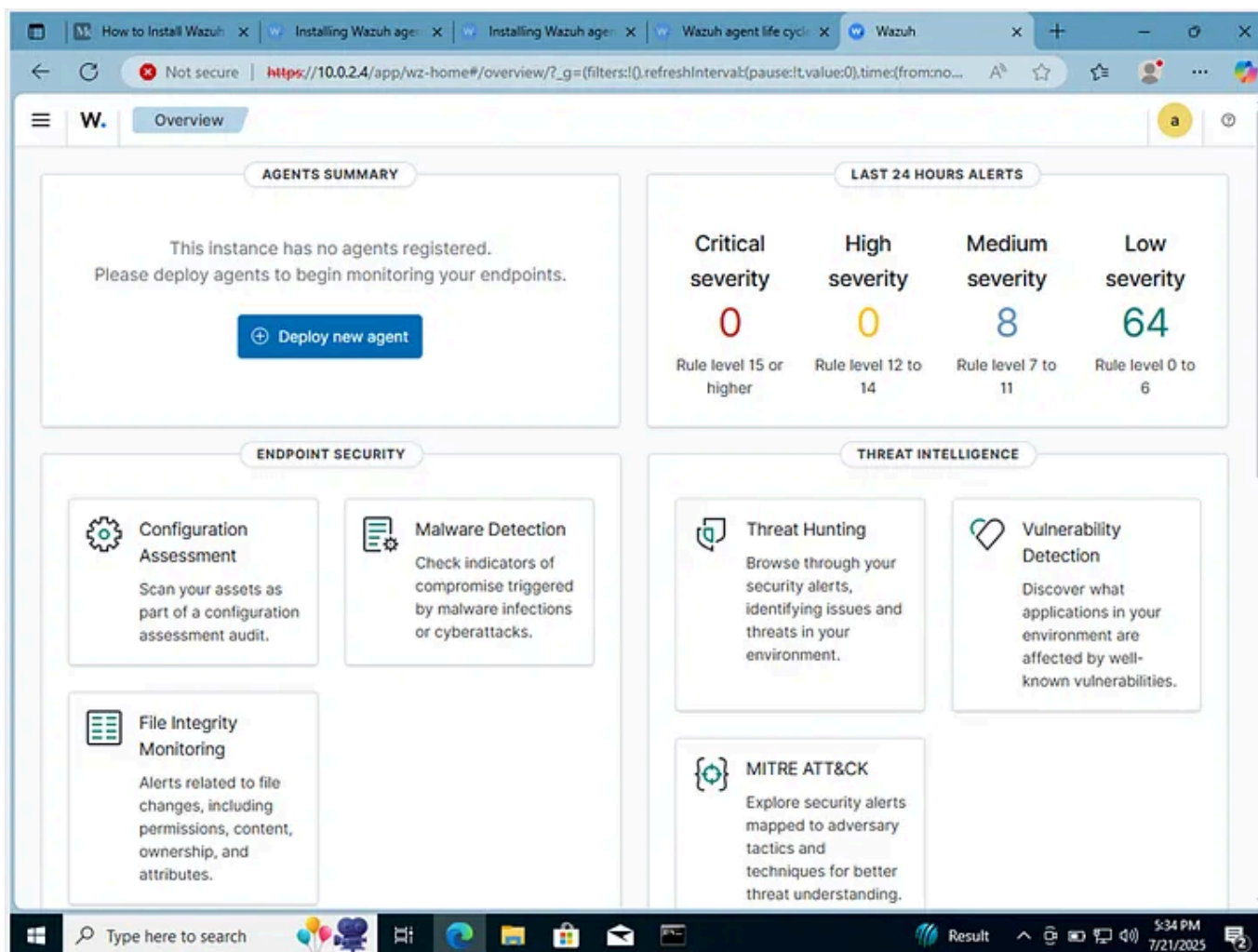


Fig. 16

Navigating the Wazuh Server

At this point, I will SSH into the Wazuh server from the host system as a matter of personal preference. The Wazuh server's console does not support copy and paste, a functionality I prefer to have in my CLI. To enable SSH access, I will activate Adapter 2 in VirtualBox and configure it as Host-Only. This creates a dedicated private network between the host and the VM, allowing the host system to communicate directly with the Wazuh server without internet dependency, ensuring a stable connection for SSH sessions. To learn more about the different network modes in VirtualBox, check [here](#).

The Wazuh VM needs to be turned off before applying this setting and restarted afterward.

To configure Adapter 2 (Fig. 17):

- Go to **Settings > Network > Adapter 2**
- Check **Enable Network Adapter**
- Select **Host-Only Adapter** from the drop-down menu
- Click **OK** to apply the changes

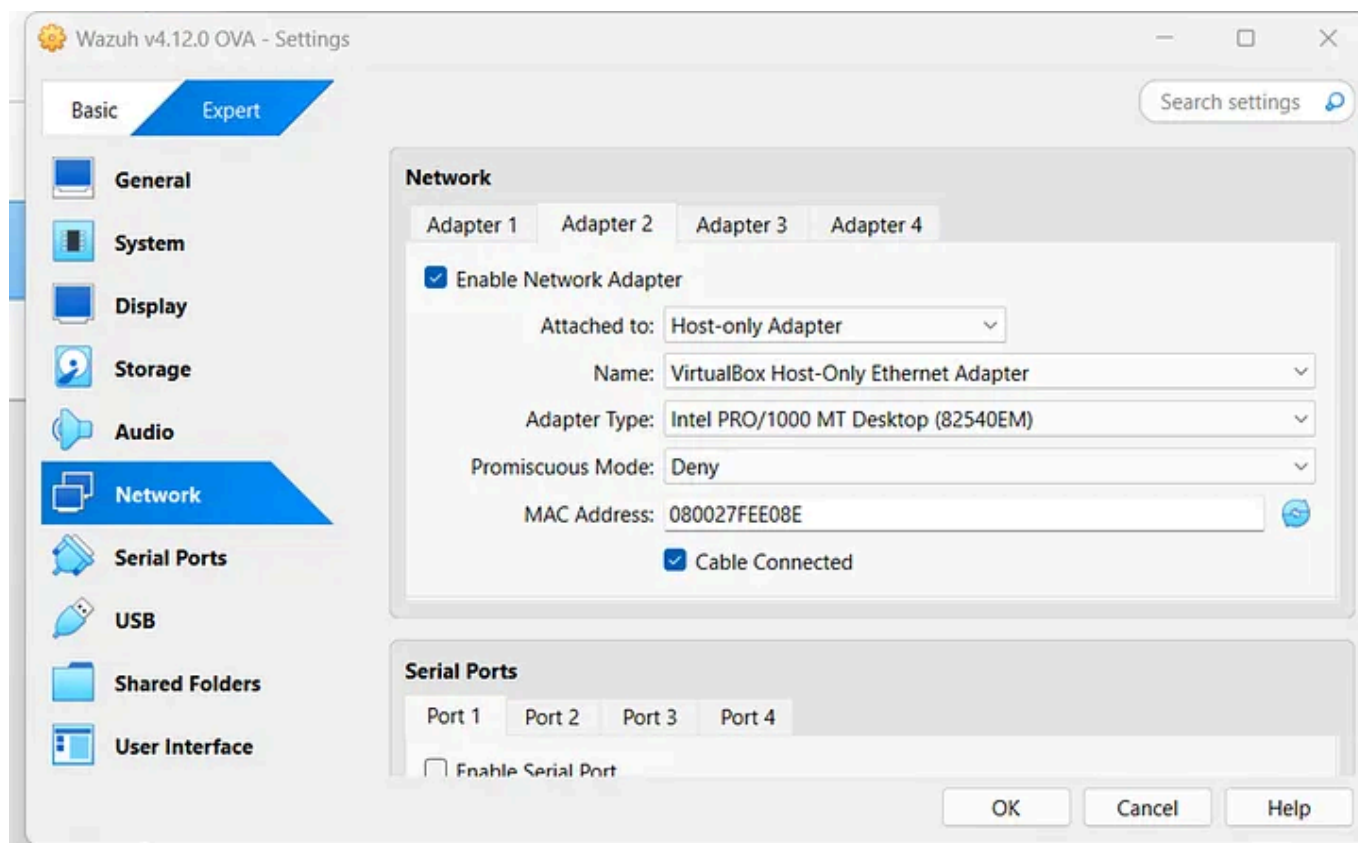


Fig. 17

To SSH into the wazuh server from powershell, first you need to check if SSH is running in the wazuh VM.

Check using:

```
sudo systemctl status sshd
```

Install SSH (if not already running):

```
sudo yum install -y openssh-server
```

Start up the SSH service:

```
sudo systemctl start sshd
```

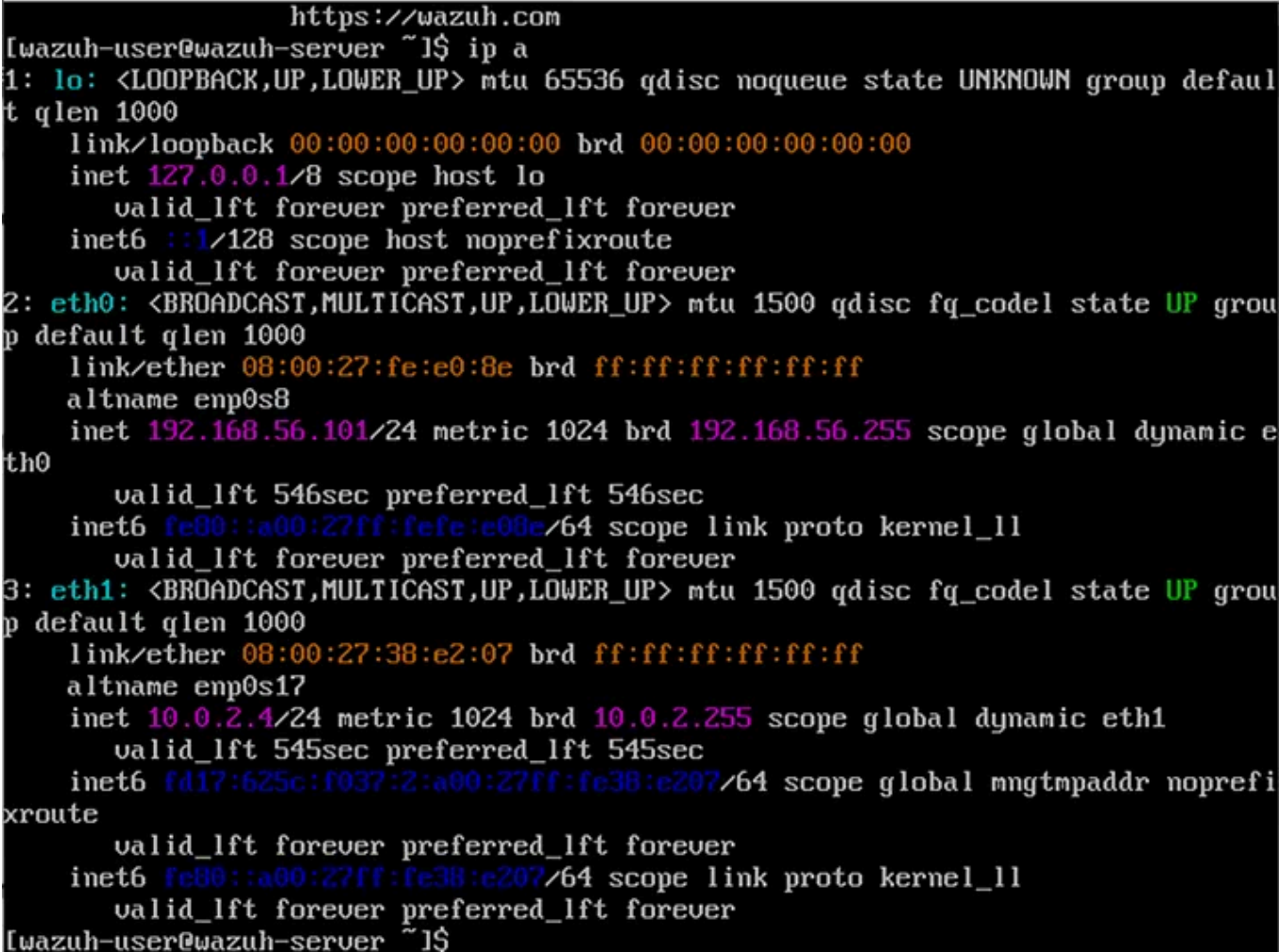
Check the status again to confirm it is running.

```
sudo systemctl status sshd
```

SSH using windows powershell in your host by running the command:

```
ssh wazuh-user@192.168.156.101
```

Make sure to use the IP address associated with the host only adapter in your case. This is the IP address associated with `inet` under `eth0` (you can check using the `ip a` command (Fig. 18).



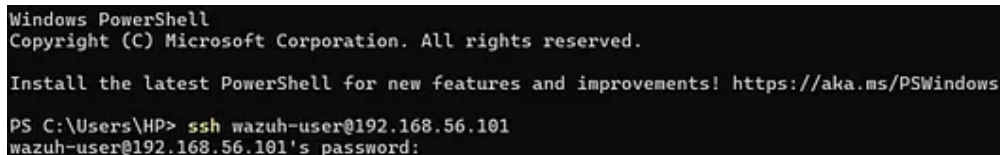
```

https://wazuh.com
[wazuh-user@wazuh-server ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fe:e0:8e brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.56.101/24 metric 1024 brd 192.168.56.255 scope global dynamic eth0
        valid_lft 546sec preferred_lft 546sec
    inet6 fe80::a00:27ff:fe38:e207/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:38:e2:07 brd ff:ff:ff:ff:ff:ff
    altname enp0s17
    inet 10.0.2.4/24 metric 1024 brd 10.0.2.255 scope global dynamic eth1
        valid_lft 545sec preferred_lft 545sec
    inet6 fd17:625c:f037:2:a00:27ff:fe38:e207/64 scope global mngtmpaddr noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe38:e207/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
[wazuh-user@wazuh-server ~]#
  
```

Fig. 18

You will be prompted to enter password after running `ssh wazuh-user@192.168.156.101`

Enter the default password of the wazuh server, and voilà, you are now connected to the server via SSH (Fig. 19).



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP> ssh wazuh-user@192.168.56.101
wazuh-user@192.168.56.101's password:
```

Fig. 19

The first computer bug was a literal bug. In 1947, engineers working on the Harvard Mark II found a moth trapped in a relay, causing a malfunction.

Step 3: INSTALLING WAZUH AGENT.

To start the installation process, on the Windows VM, download the Windows installer from the official Wazuh website [here](#). You can complete the installation using either the CLI or the GUI. I'll be using the GUI.

Run the Windows installer and follow the prompts in the installation wizard to complete the setup.

Read and accept the terms and conditions to proceed with the installation. Wait for the installation to complete (Fig. 20).

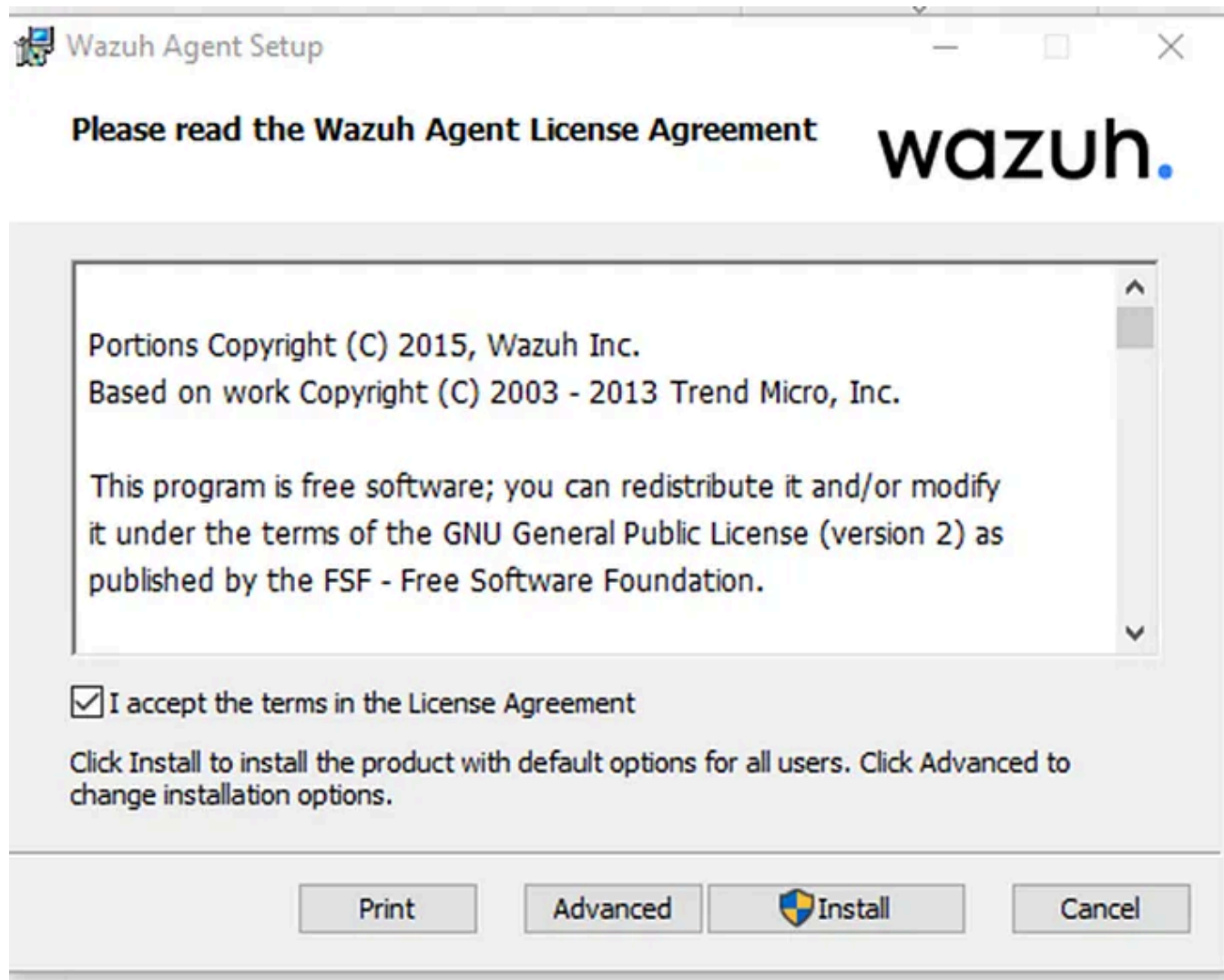


Fig. 20

Then check **Run Agent configuration interface** to launch the **Wazuh Agent Manager** (Fig. 21).

From here, you can link the agent to the server, as well as start, stop, or restart the agent.

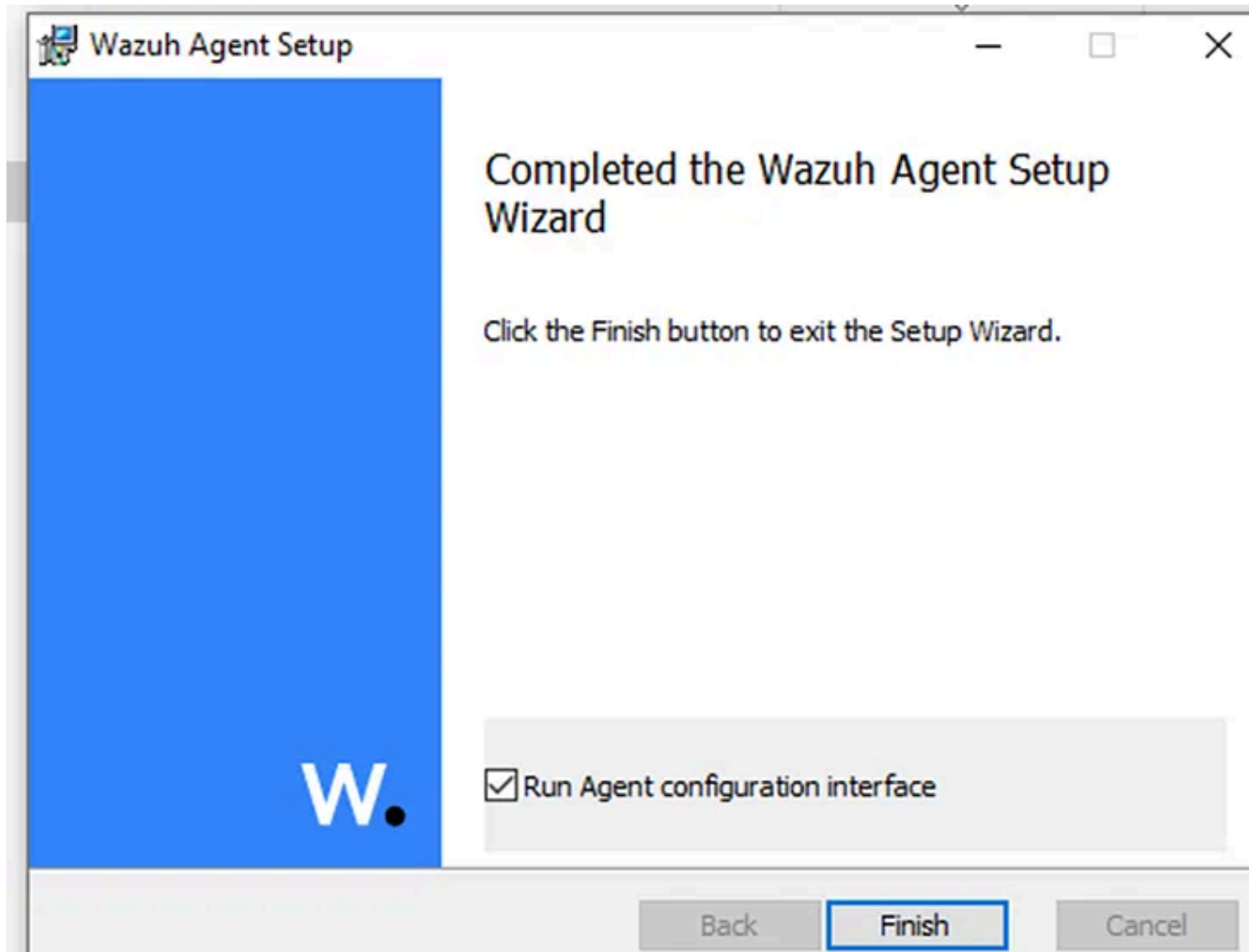


Fig. 21

After clicking on **Finish**, the Wazuh Agent Manager loads (Fig. 22).

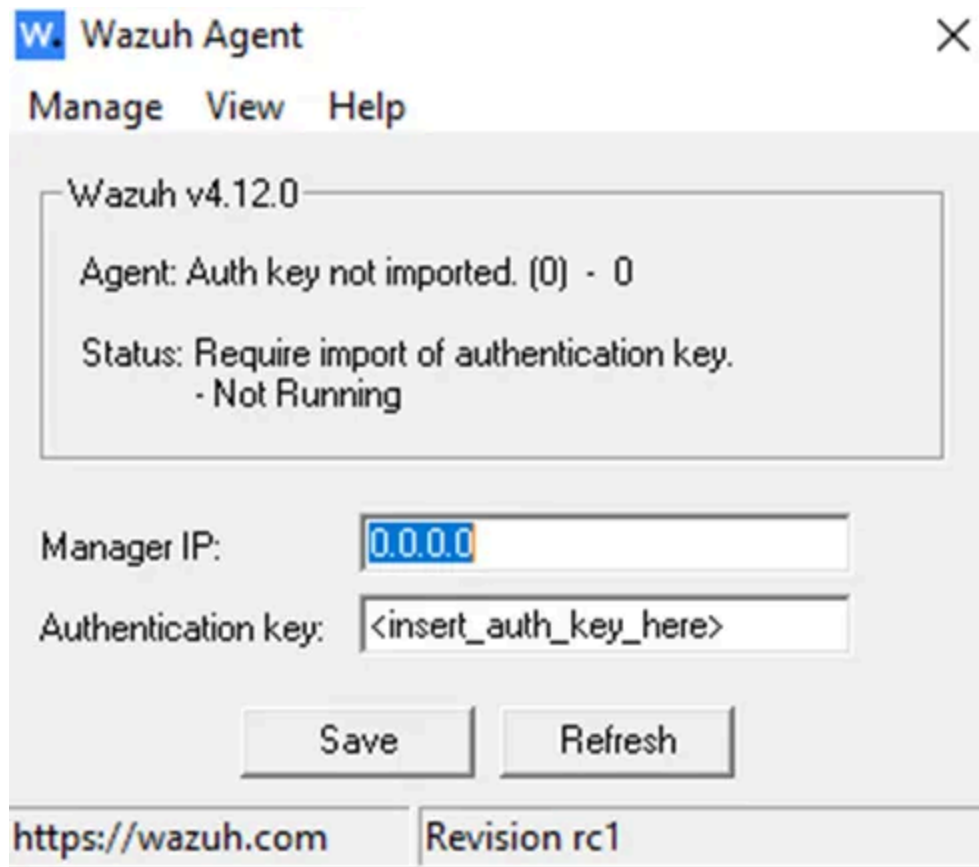


Fig. 22

In the **Wazuh Agent Manager** on Windows, enter the **Wazuh server IP address**.

To get the authentication key, you first need to add the agent on the Wazuh manager. On the Wazuh VM terminal, run (Fig. 23):

```
sudo /var/ossec/bin/manage_agents
```

- Press A to Add an agent
- Enter:

- Name: for example, agent1
- IP address: the IP address of the Windows VM running the Wazuh agent (you can find this by running ipconfig in the Windows Command Prompt)

IP address: the IP address of the Windows VM running the Wazuh agent (you can find this by running ipconfig in the Windows Command Prompt)

- An ID will be automatically assigned (in this example, 001)
- After adding the agent, press E to Extract key
- Select the agent you just added
- A long base64-encoded key will be displayed, for example:

```
MDAxIDAwMSB1YnVudHUgMTkyLjE2OC41Ni4xMDEgMDBmMWI2YjYzYmU0ZDI1MDgzNmI4NzM1MWNkYzFi
```



```

[wazuh-user@wazuh-server ~]$ sudo /var/ossec/bin/manage_agents

*****
* Wazuh v4.12.0 Agent manager.          *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: A

- Adding a new agent (use '\q' to return to the main menu).
Please provide the following:
  * A name for the new agent: agent1
  * The IP Address of the new agent: 10.0.2.5
Confirm adding it?(y/n): y
Agent added with ID 001.

*****
* Wazuh v4.12.0 Agent manager.          *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: E

Available agents:
  ID: 001, Name: agent1, IP: 10.0.2.5
Provide the ID of the agent to extract the key (or '\q' to quit): 001

Agent key information for '001' is:
MDAxIGFnZW50MSAxMCA4wLjIuNSA5YzZkZjNlZTRiODAwMzI2YmJkZWZjM2Y3ODAwOTlmMTk0MmYyZmYyNTY2Y2U1ZmM4OWQ1MzhhYWw0MGEzOTc5ZjA0

** Press ENTER to return to the main menu.

```

Fig. 23

Next, paste the full key you extracted from the manager and click Save (Fig. 24, 25).

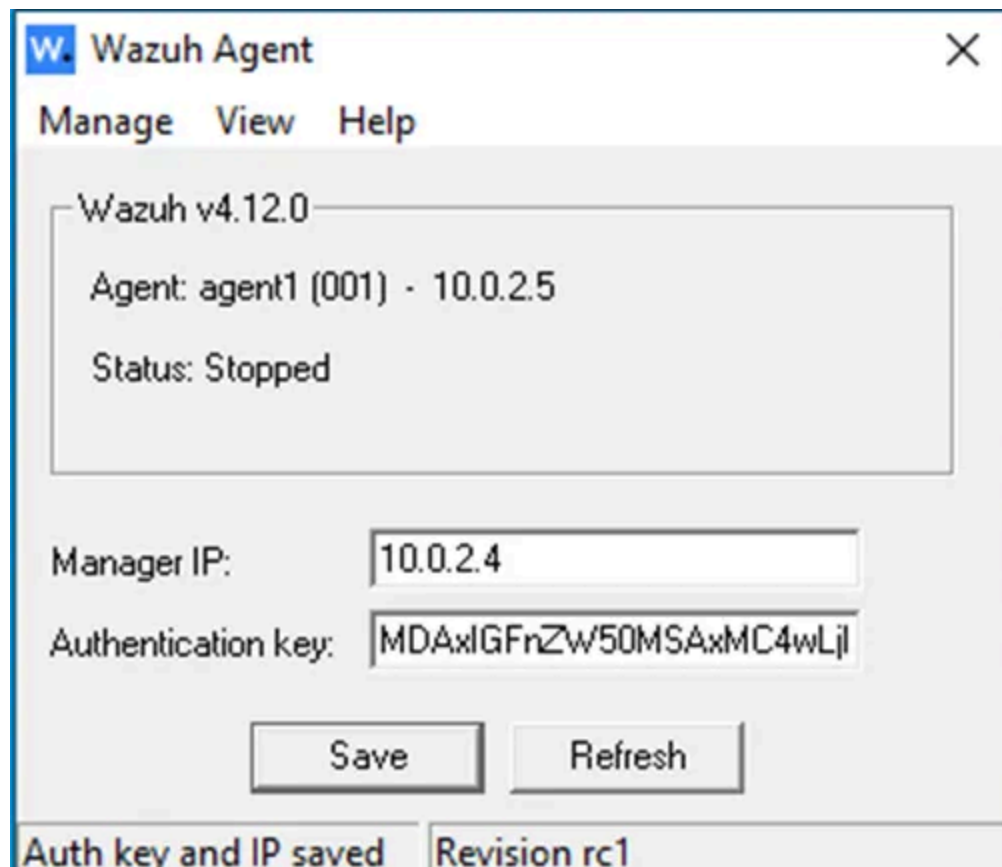


Fig. 24

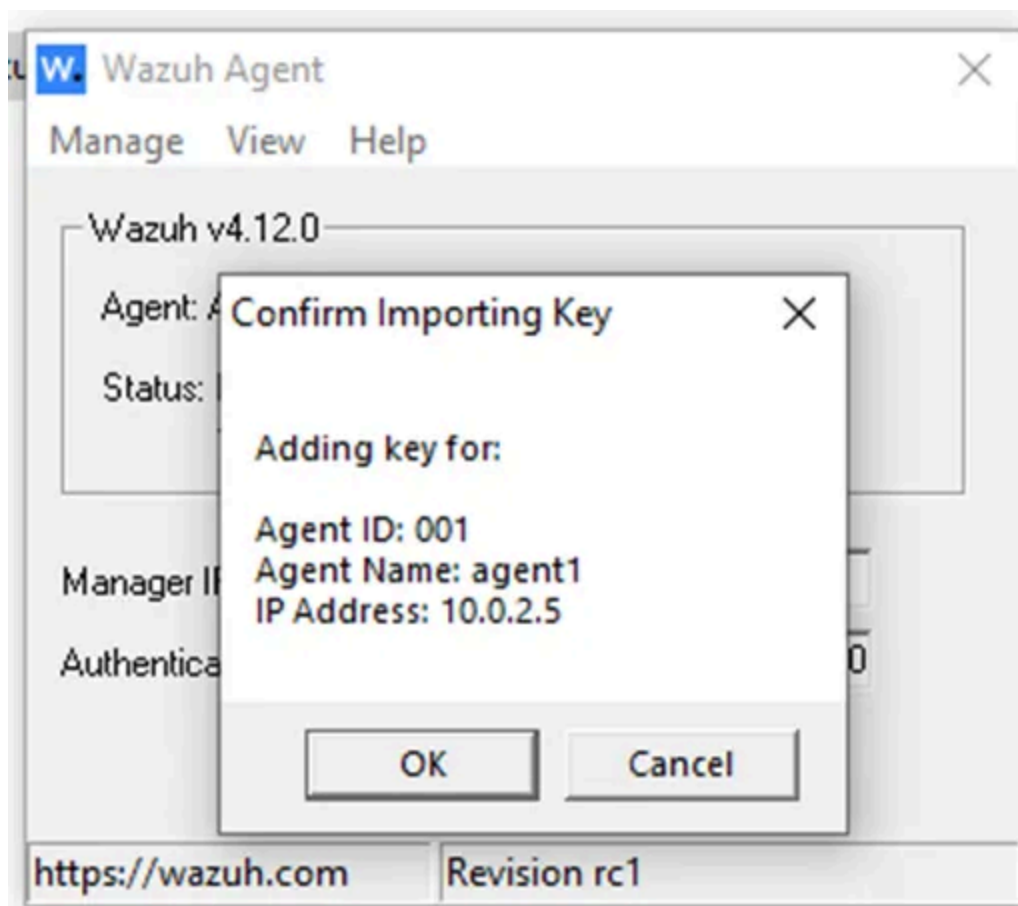


Fig. 25

Go to **Manage**, then **Start** the agent (Fig. 26).

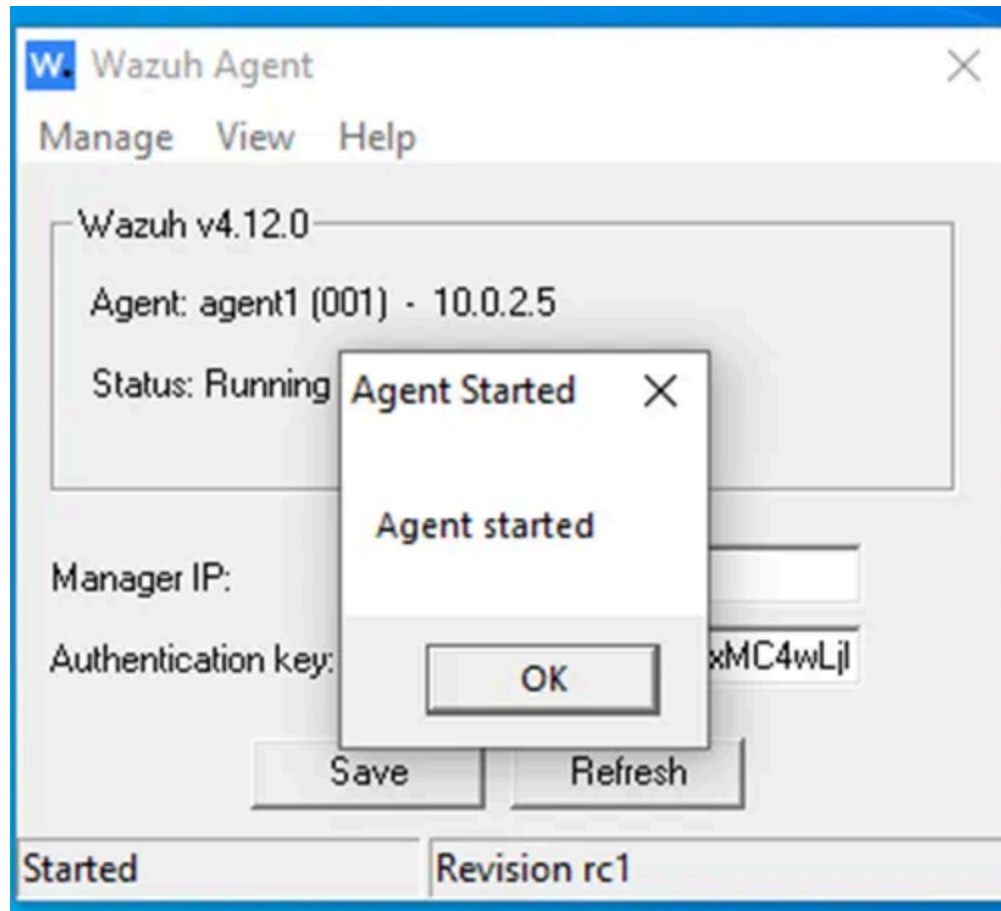


Fig. 26

You can verify it's connected by checking the **Wazuh dashboard > Agents** (Fig. 27).

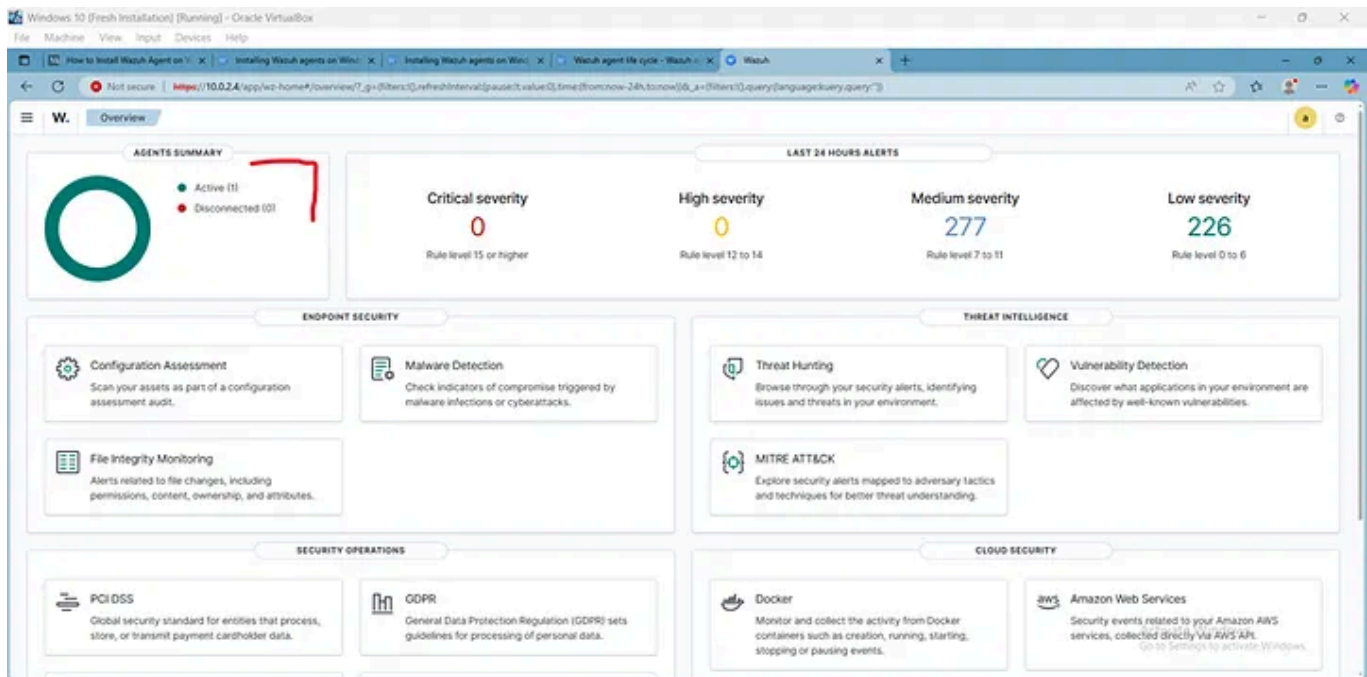


Fig. 27

At this point, I have installed the Wazuh server, accessed the dashboard through a browser, and deployed the Wazuh agent on a Windows endpoint. If you've followed the steps, you should be able to do the same.

Cheers!

Detection Engineering

Homelab

Wazuh

Threat Detection

Open Source



Written by Frederick Adigun

3 followers · 2 following

Edit profile