Integrated Network & Host Monitoring for Enterprise Web Infrastructure: A VM-based SOC Lab(Guide)

The steps implemented in this Guide:

1.Different OS setup in the VM (kali-linux,ubuntu,windows 10, windows server,linux-server,cloud server).

2.Making the Central SIEM Manger into Bridge Mode for Network logs(Ubuntu).

3.Seting Up the Network based IDS(snort) on Central SIEM Manger(Ubuntu).

4.In the Central SIEM Manger setup wazuh-Manger(Ubuntu).

5.Setup of wazuh in the different OS to forward the logs (kali-linux,window-10,windows-server,cloud serve,linux-server).

6. Make detection rules in the central SIEM .

7.Simluating the attacks to check the detection rules are working and making a ticket in the slack

Step-1:

Different OS setup in the VM (kali-linux,ubuntu,windows 10, windows server,linux-server,cloud server).

- Kali-Linux os (https://www.kali.org/get-kali/#kali-virtual-machines)
- Ubuntu OS (https://ubuntu.com/download/desktop)
- Windows-10 os(https://www.microsoft.com/en-in/software-download/windows10)
- Windows-server 2022 (https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2022)
- Kali-linux server os (https://www.kali.org/get-kali/#kali-virtual-machines)

Youtube Guide

- **How to Create a Windows 11 Virtual Machine With VMware YouTube**
- **How to install Ubuntu Linux (Desktop) in VMware Workstation YouTube**
- **Installing Kali on VMware YouTube**

2.How To Enable Promiscuous Mode In Vmware Workstation(Ubuntu)

Promiscuous mode is a network interface card (NIC) mode where the card passes all traffic it receives to the CPU rather than only frames addressed to it. This is commonly used in network analysis and monitoring applications, allowing tools to inspect all traffic on the network segment. In VMware Workstation, enabling promiscuous mode can be essential for certain tests, penetration testing practices, use of network sniffers, or learning environments involving network traffic analysis. This detailed guide provides step-by-step instructions on enabling promiscuous mode in VMware Workstation.

Step-by-Step Guide: Enabling Promiscuous Mode

Step 1: Accessing VM Settings

1. **Open VMware Workstation:** Launch the VMware Workstation application on your host machine.
2. **Select the Virtual Machine:** In the VMware Workstation interface, click on the virtual machine for which you want to enable promiscuous mode.
3. **Power Off the VM:** If the VM is currently running, you must power it off first. Promiscuous mode cannot be enabled while the VM is active.

4. **Open Settings:** Right-click on the VM and select “Settings,” or from the VM menu, click on “Edit Virtual Machine Settings.”

Step 2: Configuring the Network Adapter

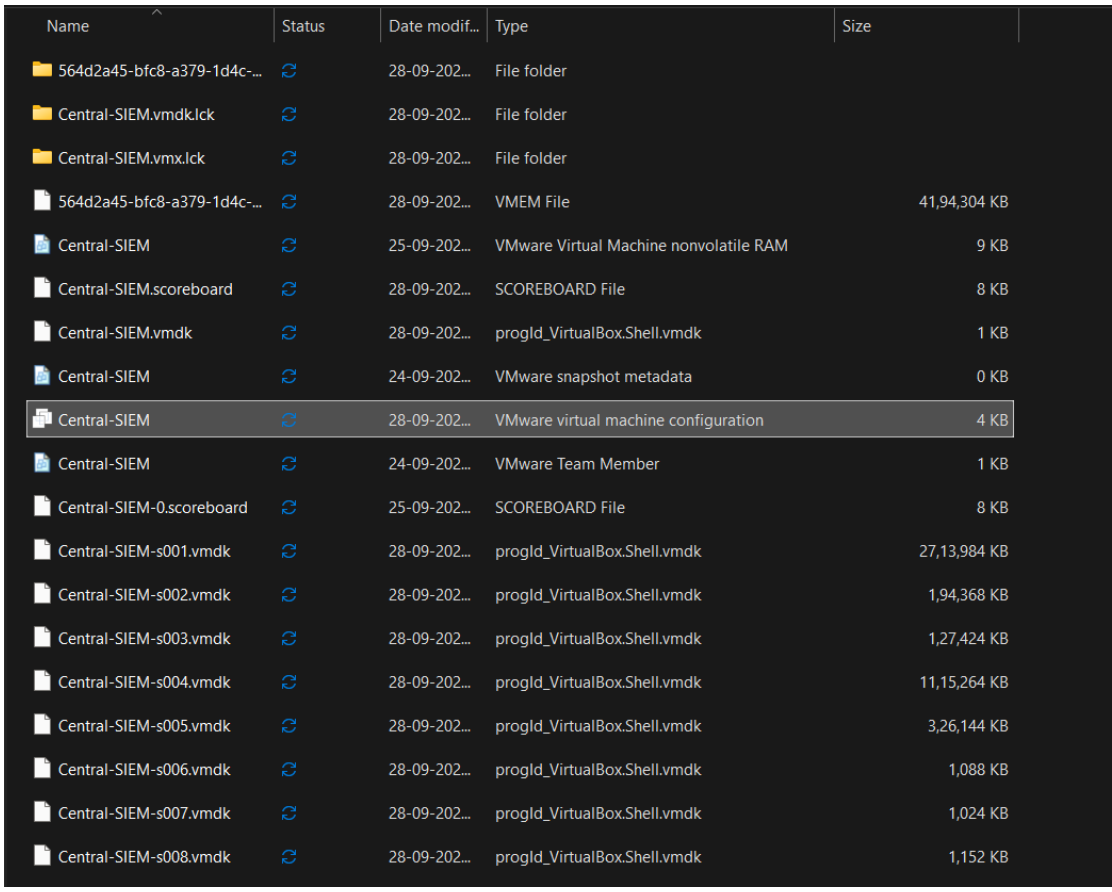
1. **Select the Network Adapter:** In the virtual machine settings window, find and click on the “Network Adapter” option. Here, you will see settings related to the network for the virtual machine.
2. **Change Network Connection Type:** Ensure that the network connection type is set appropriately:
    - For promiscuous mode to take effect, the virtual network adapter should typically be set to "Bridged" or "Host-only" mode.
    - If set to "NAT," promiscuous mode will not work effectively since NAT configurations isolate VM traffic.
3. **Enable Promiscuous Mode:** Depending on your version of VMware Workstation, you may have an option labeled "Promiscuous Mode" within the settings of the network adapter:
    - If this option is available, slide the control to "Allow All" or "Allow VMs," depending on your needs:
        - **Allow All:** All traffic to and from the network adapter will be captured.
        - **Allow VMs:** Only traffic to and from VMs running on the same host will be captured.

- If the option is not directly available, we will manually modify the configuration files in the next steps.

Step 3: Editing The VMX File (if applicable)

1. **Locate the VM's .vmx File:**

   The .vmx file contains configuration settings for your virtual machine. To locate it, open the file system, navigate to your virtual machine's directory, and find the .vmx file corresponding to the VM you are configuring.



Name	Status	Date modif...	Type	Size
564d2a45-bfc8-a379-1d4c-...		28-09-202...	File folder	
Central-SIEM.vmdk.lck		28-09-202...	File folder	
Central-SIEM.vmx.lck		28-09-202...	File folder	
564d2a45-bfc8-a379-1d4c-...		28-09-202...	VMEM File	41,94,304 KB
Central-SIEM		25-09-202...	VMware Virtual Machine nonvolatile RAM	9 KB
Central-SIEM.scoreboard		28-09-202...	SCOREBOARD File	8 KB
Central-SIEM.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	1 KB
Central-SIEM		24-09-202...	VMware snapshot metadata	0 KB
Central-SIEM		28-09-202...	VMware virtual machine configuration	4 KB
Central-SIEM		24-09-202...	VMware Team Member	1 KB
Central-SIEM-0.scoreboard		25-09-202...	SCOREBOARD File	8 KB
Central-SIEM-s001.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	27,13,984 KB
Central-SIEM-s002.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	1,94,368 KB
Central-SIEM-s003.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	1,27,424 KB
Central-SIEM-s004.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	11,15,264 KB
Central-SIEM-s005.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	3,26,144 KB
Central-SIEM-s006.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	1,088 KB
Central-SIEM-s007.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	1,024 KB
Central-SIEM-s008.vmdk		28-09-202...	progId_VirtualBox.Shell.vmdk	1,152 KB

2. **Edit the .vmx File:**

   - Open the .vmx file in a text editor (you may need to run the editor as an administrator).

3. **Add the Promiscuous Mode Setting:**

- Add the following line to the bottom of the file:

ethernet0.promiscuous = "allow"

```
File    Edit    View
vmotion.svga.logicBlendOps = "0"
vmotion.svga.maxForcedSampleCount = "16"
vmotion.svga.gl43 = "1"
ethernet0.generatedAddress = "00:0c:29:9b:bc:57"
ethernet0.generatedAddressOffset = "0"
vmci0.id = "-576996265"
monitor.phys_bits_used = "45"
cleanShutdown = "FALSE"
softPowerOff = "FALSE"
usb:1.speed = "2"
usb:1.present = "TRUE"
usb:1.deviceType = "hub"
usb:1.port = "1"
usb:1.parent = "-1"
usb:0.present = "TRUE"
usb:0.deviceType = "hid"
usb:0.port = "0"
usb:0.parent = "-1"
ethernet0.promiscuous = "allow"
```

- This line instructs VMware to allow promiscuous mode for the first network adapter. If you have multiple network adapters, you may want to refer to them as ethernet1, ethernet2, etc.

Optional Verification via Command Line

You can also check if your interface is in promiscuous mode:

ip link show eth0(here your NIC name)

Look for the word PROMISC in the output. Example:

2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> ...

- PROMISC → Promiscuous mode is active.
- Missing → It is not enabled.

Force the interface into promiscuous mode

Run:

sudo ip link set eth0 promisc on

```
fazal@fazal-sec:~$ ip link show ens33
2: ens33: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 100
0
    link/ether 00:0c:29:9b:bc:57 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
fazal@fazal-sec:~$
```

3.Seting Up the Network based IDS(snort) on Central SIEM Manger(Ubuntu).

1. Install Snort on Ubuntu

Run these commands as root or with sudo:

sudo apt update && sudo apt upgrade -y

sudo apt install snort -y

During installation, it will ask for **network interface** → choose the interface you want to monitor (e.g., eth0).

👉 Check Snort version:

snort -V

```
sec@sec:~$ snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.4 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.3

sec@sec:~$
```

Test Snort in IDS Mode

Run with default rules:

sudo snort -A console -q -c /etc/snort/snort.conf -i eth0

- -A console → shows alerts on screen
- -q → quiet mode
- -i eth0 → interface

```
root@sec:/# nano /etc/snort/snort.conf
root@sec:/# snort -A console -q -c /etc/snort/snort.conf -i ens33
09/29-17:08:24.667672  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:08:32.041319  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:08:37.979604  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:08:42.690935  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:08:42.972356  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.31.42:48658 -> 192.168.31.102:705
09/29-17:08:43.025015  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.31.42:48658 -> 192.168.31.102:161
09/29-17:08:49.039112  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:08:55.592040  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:00.711430  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:01.941226  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:03.169917  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:04.602523  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:06.431521  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.31.42:57875 -> 192.168.31.102:161
09/29-17:09:06.511085  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.31.42:57875 -> 192.168.31.102:705
09/29-17:09:07.675809  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:12.590425  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:13.003269  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:18.940286  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:25.493370  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:31.021205  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:36.551270  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:44.331520  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:50.271127  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:09:55.799321  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:10:01.328942  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:10:07.472346  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:10:12.592210  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:10:17.191024  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.31.42:51292 -> 192.168.31.102:705
09/29-17:10:17.252133  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.31.42:51292 -> 192.168.31.102:161
09/29-17:10:18.531741  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
09/29-17:10:26.108085  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::16
```

Check for video tutorial

[snort-setup]

4.In the Central SIEM Manger setup the ELK stack and wazuh-Manger(Ubuntu).

1.Wazuh manger setup (Do paste this steps according order)

- Sudo apt install curl

```
root@sec:/# sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libgl1-amber-dri libglapi-mesa libllvm19
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 341 not upgraded.
Need to get 226 kB of archives.
After this operation, 534 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.6 [226 kB]
Fetched 226 kB in 3s (72.5 kB/s)
Selecting previously unselected package curl.
(Reading database ... 151385 files and directories currently installed.)
Preparing to unpack .../curl_8.5.0-2ubuntu10.6_amd64.deb ...
Unpacking curl (8.5.0-2ubuntu10.6) ...
Setting up curl (8.5.0-2ubuntu10.6) ...
Processing triggers for man-db (2.12.0-4build2) ...
```

- Sudo apt install default-jdk -y

```
root@sec:/# sudo apt install default-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libgl1-amber-dri libglapi-mesa libllvm19
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ca-certificates-java default-jdk-headless default-jre default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni
  libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-21-jdk openjdk-21-jdk-headless openjdk-21-jre openjd
  xorg-sgml-doctools xtrans-dev
Suggested packages:
  libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-21-demo openjdk-21-source visualvm fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-m
  fonts-indic
The following NEW packages will be installed:
  ca-certificates-java default-jdk default-jdk-headless default-jre default-jre-headless fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrap
  libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-21-jdk openjdk-21-jdk-headless openjdk-21-jre openjd
  xorg-sgml-doctools xtrans-dev
0 upgraded, 24 newly installed, 0 to remove and 341 not upgraded.
Need to get 135 MB of archives.
After this operation, 316 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble/main amd64 ca-certificates-java all 20240118 [11.6 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu noble/main amd64 java-common all 0.75+exp1 [6,798 B]
Get:3 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 openjdk-21-jre-headless amd64 21.0.8+9~us1-0ubuntu1~24.04.1 [46.4 MB]
Get:4 http://in.archive.ubuntu.com/ubuntu noble/main amd64 default-jre-headless amd64 2:1.21-75+exp1 [3,094 B]
Get:5 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 openjdk-21-jre amd64 21.0.8+9~us1-0ubuntu1~24.04.1 [228 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu noble/main amd64 default-jre amd64 2:1.21-75+exp1 [922 B]
Get:7 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 openjdk-21-jdk-headless amd64 21.0.8+9~us1-0ubuntu1~24.04.1 [82.7 MB]
Get:8 http://in.archive.ubuntu.com/ubuntu noble/main amd64 default-jdk-headless amd64 2:1.21-75+exp1 [960 B]
Get:9 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 openjdk-21-jdk amd64 21.0.8+9~us1-0ubuntu1~24.04.1 [1,645 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu noble/main amd64 default-jdk amd64 2:1.21-75+exp1 [926 B]
Get:11 http://in.archive.ubuntu.com/ubuntu noble/main amd64 fonts-dejavu-extra all 2.37-8 [1,947 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu noble/main amd64 libatk-wrapper-java all 0.40.0-3build2 [54.3 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu noble/main amd64 libatk-wrapper-java-jni amd64 0.40.0-3build2 [46.4 kB]
```

- (curl -sO https://packages.wazuh.com/4.13/wazuh-install.sh)
- bash wazuh-install.sh -a

```
29/09/2025 18:07:40 INFO: Generating Wazuh dashboard certificates.
29/09/2025 18:07:41 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
29/09/2025 18:07:41 INFO: --- Wazuh indexer ---
29/09/2025 18:07:41 INFO: Starting Wazuh indexer installation.
29/09/2025 18:16:02 INFO: Wazuh indexer installation finished.
29/09/2025 18:16:07 INFO: Wazuh indexer post-install configuration finished.
29/09/2025 18:16:07 INFO: Starting service wazuh-indexer.
29/09/2025 18:16:38 INFO: wazuh-indexer service started.
29/09/2025 18:16:38 INFO: Initializing Wazuh indexer cluster security settings.
29/09/2025 18:16:45 INFO: Wazuh indexer cluster security configuration initialized.
29/09/2025 18:16:45 INFO: Wazuh indexer cluster initialized.
29/09/2025 18:16:45 INFO: --- Wazuh server ---
29/09/2025 18:16:45 INFO: Starting the Wazuh manager installation.
29/09/2025 18:21:02 INFO: Wazuh manager installation finished.
29/09/2025 18:21:02 INFO: Wazuh manager vulnerability detection configuration finished.
29/09/2025 18:21:02 INFO: Starting service wazuh-manager.
29/09/2025 18:21:20 INFO: wazuh-manager service started.
29/09/2025 18:21:20 INFO: Starting Filebeat installation.
29/09/2025 18:21:39 INFO: Filebeat installation finished.
29/09/2025 18:21:52 INFO: Filebeat post-install configuration finished.
29/09/2025 18:21:52 INFO: Starting service filebeat.
29/09/2025 18:21:54 INFO: filebeat service started.
29/09/2025 18:21:54 INFO: --- Wazuh dashboard ---
29/09/2025 18:21:54 INFO: Starting Wazuh dashboard installation.
29/09/2025 18:26:50 INFO: Wazuh dashboard installation finished.
29/09/2025 18:26:50 INFO: Wazuh dashboard post-install configuration finished.
29/09/2025 18:26:50 INFO: Starting service wazuh-dashboard.
29/09/2025 18:26:52 INFO: wazuh-dashboard service started.
29/09/2025 18:26:54 INFO: Updating the internal users.
29/09/2025 18:27:58 INFO: A backup of the internal users has been saved in the /etc/wazuh-indexer/internalusers-backup folder.
29/09/2025 18:28:42 INFO: The filebeat.yml file has been updated to use the Filebeat Keystore username and password.
29/09/2025 18:29:27 INFO: Initializing Wazuh dashboard web application.
29/09/2025 18:29:28 INFO: Wazuh dashboard web application initialized.
29/09/2025 18:29:28 INFO: --- Summary ---
```

Starting of the Wazuh-magner

- **sudo systemctl start wazuh-manager**
- **sudo systemctl start filebeat**
- **sudo systemctl start wazuh-dashboard**
- **sudo systemctl start wazuh-indexer**

```
sec@sec:~$ sudo systemctl start wazuh-manager
[sudo] password for sec:
sec@sec:~$ sudo systemctl start filebeat
sec@sec:~$ sudo systemctl start wazuh-dashboard
sec@sec:~$ sudo systemctl start wazuh-indexer
sec@sec:~$ sudo systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; pr>
     Active: active (running) since Wed 2025-10-01 23:55:06 IST; 1 day 16h ago
      Tasks: 201 (limit: 4545)
     Memory: 690.1M (peak: 1.6G swap: 258.1M swap peak: 321.1M)
        CPU: 33min 9.795s
     CGroup: /system.slice/wazuh-manager.service
             ├─12893 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
             ├─12894 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
             ├─12895 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
             ├─12898 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
             ├─12901 /var/ossec/framework/python/bin/python3 /var/ossec/api/scr>
             ├─12942 /var/ossec/bin/wazuh-authd
             ├─12959 /var/ossec/bin/wazuh-db
             ├─13134 /var/ossec/bin/wazuh-execd
             ├─13160 /var/ossec/bin/wazuh-analysisd
             ├─13205 /var/ossec/bin/wazuh-syscheckd
             ├─13222 /var/ossec/bin/wazuh-remoted
             ├─13258 /var/ossec/bin/wazuh-logcollector
             ├─13277 /var/ossec/bin/wazuh-monitord
             └─13419 /var/ossec/bin/wazuh-modulesd

Oct 01 23:55:03 sec env[13283]: wazuh-syscheckd already running...
lines 1-23...skipping...
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-10-01 23:55:06 IST; 1 day 16h ago
      Tasks: 201 (limit: 4545)
```

Setup of the Wazuh Agent in different OS

- In Wazuh manger(Central SIEM) in the terminal add this command (sudo /var/ossec/bin/manage_agents)

```
sec@sec:~$ sudo /var/ossec/bin/manage_agents
[sudo] password for sec:


****************************************
* Wazuh v4.13.1 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: linux_web_server
   * The IP Address of the new agent: 192.168.31.32
```

- Next in the manger of copy the agent key

```
sec@sec:~$ sudo /var/ossec/bin/manage_agents


****************************************
* Wazuh v4.13.1 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: kali_machine, IP: any
   ID: 002, Name: windows_10, IP: 192.168.31.95
   ID: 003, Name: linux_web_server, IP: 192.168.31.121
   ID: 004, Name: windows_server, IP: 192.168.31.167
   ID: 006, Name: EC2_web_server, IP: 100.93.244.83
Provide the ID of the agent to extract the key (or '\q' to quit): 001

Agent key information for '001' is:
MDAxIGthbGlfbWFjaGluZSBhbnkgZTRmOWFiMTdhNTA4Mzc1MDA2NDAyNTJiMmNjNTdkMmYxOWI3M2RkYzU4NWQ5MDQ1YmNiYjA4NTI1NDBmNTI1Nw==

** Press ENTER to return to the main menu.
```

- In the OS want to add the wazuh agent (**Configure the Wazuh Agent as Edit the agent config file**
- **sudo nano /var/ossec/etc/ossec.conf**

```
  GNU nano 8.1
<!--
  Wazuh - Agent - Default configuration for kali 2024.3
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <client>
    <server>
      <address>192.168.31.102</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>kali, kali2024, kali2024.3</config-profile>
    <notify_time>20</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
    <enrollment>
      <enabled>yes</enabled>
      <agent_name>linux_web_server</agent_name>
      <authorization_pass_path>etc/authd.pass</authorization_pass_path>
    </enrollment>
  </client>

  <client_buffer>
    <!-- Agent buffer options -->
    <disabled>no</disabled>
    <queue_size>5000</queue_size>
    <events_per_second>500</events_per_second>
  </client_buffer>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
```

Replace MANAGER_IP with your manager's IP

Example: if your Wazuh Manager is at 192.168.1.100

Save and exit

- In nano: CTRL+O, then Enter, then CTRL+X

Restart the agent

sudo systemctl daemon-reexec

sudo systemctl restart wazuh-agent

sudo systemctl status wazuh-agent

On your **Kali (where the agent is installed)**, the logs are located in:

/var/ossec/logs/ossec.log

now the agent machine paste this command and import the agent key which (**sudo /var/ossec/bin/manage_agents)**

```
root@kali:~# sudo /var/ossec/bin/manage_agents


****************************************
* Wazuh v4.13.1 Agent manager.         *
* The following options are available: *
****************************************
   (I)mport key from the server (I).
   (Q)uit.
Choose your action: I or Q: I

* Provide the Key generated by the server.
* The best approach is to cut and paste it.
*** OBS: Do not include spaces or new lines.

Paste it here (or '\q' to quit):
```

for other type of os refer this (wazuh-guide-endpoints)

Launching an EC2 Instance with Amazon Linux

Prerequisites

1. **AWS Account** — you need an AWS account with permission to create EC2 instances.
2. **IAM Permissions** — ensure your IAM user has the necessary EC2, VPC, and KeyPair permissions.
3. **Region Selection** — choose an AWS region close to your user base to reduce latency.

Steps to Launch

1. **Open EC2 Console**
   Go to the AWS Management Console → Services → EC2. Amazon Web Services, Inc.+1
2. **Launch Instance Wizard**
   In the EC2 dashboard, click **"Launch instance"**. AWS Documentation+2AWS Documentation+2

3. **Name & Tags**
   Give your instance a descriptive name tag (e.g. web-server, dev-instance). AWS Documentation+2TechTarget+2
4. **Choose AMI (Amazon Machine Image)**
   Select Amazon Linux (Amazon Linux 2 or Amazon Linux 2023) as your OS. AWS Documentation+4AWS Documentation+4AWS Documentation+4
   - To always get the latest Amazon Linux image in scripts or CLI, you can use the SSM public parameter like /aws/service/ami-amazon-linux-latest/... AWS Documentation
5. **Choose Instance Type**
   Pick a size (for example t2.micro or t3.micro if under the free tier) depending on your workload. AWS Documentation+2AWS Documentation+2
6. **Configure Instance Details**
   - Network and Subnet (usually default VPC)
   - IAM role (if your instance needs AWS access)
   - Auto-assign Public IP (if you want it to be reachable from the Internet)
   - Advanced options like user data scripts, shutdown behavior AWS Documentation+2AWS Documentation+2
7. **Add Storage**
   By default, Amazon Linux instances come with an EBS volume (often 8 GB). Adjust size or storage type if needed. AWS Documentation+1
8. **Configure Security Group**
   Set inbound rules (for example, allow SSH on port 22, HTTP on port 80, etc.). It's a firewall layer — be restrictive by default. AWS Documentation+3Jenkins+3AWS Documentation+3
9. **Key Pair for SSH Access**
   Create a new key pair (or use an existing one) — you'll download a .pem file. This is needed to SSH into your instance.

AWS Documentation+2AWS Documentation+2
Make sure to keep your private key safe and set proper permissions (chmod 400 key.pem on Linux/Mac). AWS Documentation+1

10. **Review & Launch**
Review all configurations. If everything is acceptable, click **Launch instance** to start it. AWS Documentation+2AWS Documentation+2

11. **Connect to the Instance via SSH**
    - Go to the EC2 console → Instances → select your instance → **Connect**. AWS Documentation+1
    - Use the SSH command (on Linux/macOS) such as:
    - ssh -i /path/to/key.pem ec2-user@your-instance-public-dns

(ec2-user is the default for Amazon Linux) AWS Documentation+2AWS Documentation+2

    - On Windows, you can use tools like PuTTY (convert the .pem to .ppk first) to SSH in.
    - Before connecting, set private key permissions: chmod 400 key.pem (Linux/macOS). AWS Documentation

12. **Terminate or Stop Instance (when done)**
When you finish your work, stop or terminate the instance to avoid unnecessary charges

Reference guid youtube video(EC2 creation)

Now need to connect cloud host with manger,need to setup the VPN service in the both

Install Tailscale(VPN)

curl -fsSL https://tailscale.com/install.sh | sh
Enable and start the service:

sudo systemctl enable --now tailscaled

◈ Step 3: Generate an Auth Key (on your laptop/desktop browser)

1. Go to https://login.tailscale.com/admin/settings/keys
2. Click **Generate auth key** (choose *Reusable* if you'll need it often).
3. Copy the key (it looks like tskey-xxxxxxxxxxxxxxxx).

◈ Step 4: Authenticate EC2 instance with the key

On EC2 terminal:

sudo tailscale up --authkey tskey-xxxxxxxxxxxxxxxx

This bypasses the browser login and directly registers your EC2 instance into your Tailscale network.

◈ Step 5: Verify Tailscale is working

tailscale status

tailscale ip -4

You should see your EC2 listed with a **100.x.x.x IP**.

◈ Step 6: Connect Manager (Private Machine) also to Tailscale

On your private machine (Wazuh Manager host):

sudo tailscale up

(or use --authkey again).

Now both the **EC2 agent** and the **Wazuh Manager** are in the same VPN, reachable by their **Tailscale IPs**.

◈ Step 7: Configure Wazuh Agent (on EC2) to talk to Manager

Edit the agent config:

sudo nano /var/ossec/etc/ossec.conf

Find:

```
<client>
  <server>
    <address>MANAGER_IP</address>
    <port>1514</port>
    <protocol>tcp</protocol>
  </server>
</client>
```

👉 Replace MANAGER_IP with your **Wazuh Manager's Tailscale IP (100.x.x.x)**.

Restart agent:

sudo systemctl restart wazuh-agent

◈ Step 8: Approve the Agent on Manager

On your manager node:

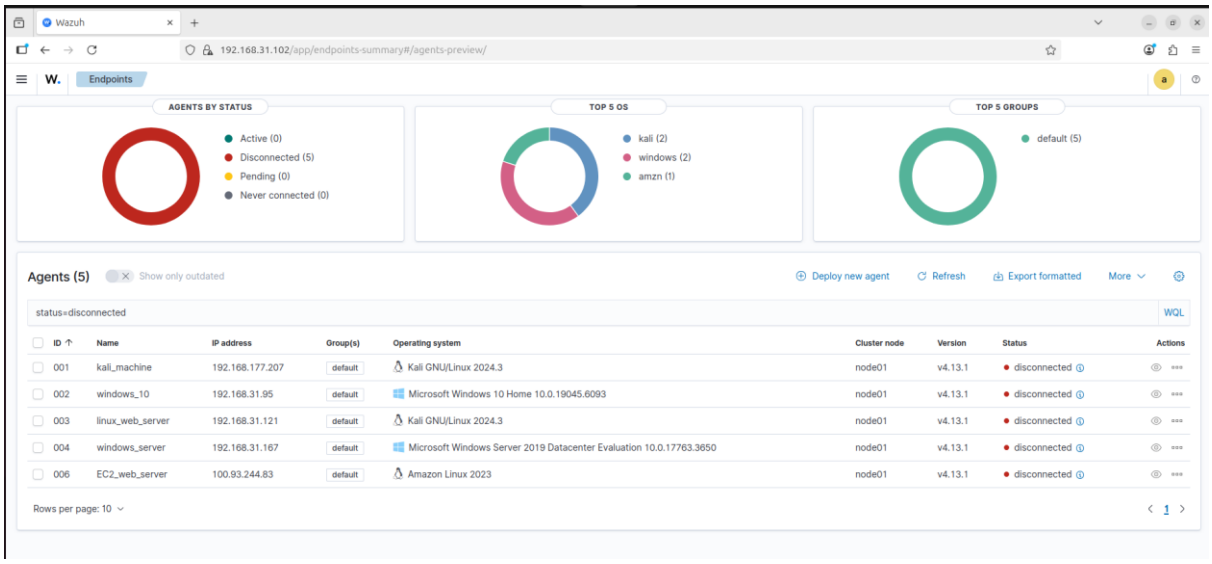
sudo /var/ossec/bin/manage_agents

- Add the new agent, copy the key.
- On EC2, import the key with:
- **sudo /var/ossec/bin/manage_agents**

→ Choose **Import key** and paste it.

Restart services on both sides.

This how wazuh dashboard looks like



6.Make detection rules in the central SIEM .

Steps to add custom alerts for all agents

1. SSH into your **Wazuh Manager**.
2. Open the local rules file:
3. **sudo nano /var/ossec/etc/rules/local_rules.xml**
4. Paste your custom rules inside <group name="local,"> ... </group> block.
   Example:
5. **<group name="local,">**
6. **<rule id="100001" level="7">**
7. **<if_sid>5712</if_sid>**
8. **<description>Multiple SSH authentication failures (possible brute force)</description>**
9. **<group>authentication, sshd, brute_force</group>**
10. **</rule>**
11. **</group>**
12. Save and exit.
13. Restart the Wazuh manager to apply changes:

Sudo systemctl restart wazuh-manager

```
GNU nano 7.2                                                    /var/ossec/etc/r

</group>

<group name="bruteforce,ssh,">
  <rule id="100101" level="10" frequency="5" timeframe="60">
    <if_matched_sid>5716</if_matched_sid>
    <description>Possible SSH brute force attack detected</description>
  </rule>
</group>

<group name="web,apache,nginx,">
  <rule id="100102" level="8">
    <match>../../etc/passwd</match>
    <description>Path traversal attempt in web request</description>
  </rule>
</group>

<group name="windows,rdp,">
  <rule id="100103" level="7">
    <if_sid>18107</if_sid>
    <description>Windows RDP login failed</description>
  </rule>
</group>


<group name="critical,">
  <rule id="100104" level="15">
    <match>root access granted</match>
    <description>CRITICAL: Unauthorized root access granted!</description>
  </rule>
</group>
```

7.Simluating the attacks to check the detection rules are working and making a ticket in the slack

webhook setup of the Slack for wazuh

1.Go to the slack and create a channel(slack)

2.Now go the slack api setup (slack-api)

3.create new app and add that channel to receive the alerts

4.In features field in left hand side bar go to incoming webhooks(slack-webhook)

5. On toggle on the right hand side ,now scroll down and click on the **add new webhook** button.

6. In the wazuh-manager go this file (/var/ossec/etc/ossec.conf)

Then paste this lines

<integration>

<name>slack</name>

<hook_url>https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK</hook_url>

<level>10</level>

<alert_format>json</alert_format>

</integration>

At hookurl paste your webhook url

```
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>15m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>


  <integration>
   <name>slack</name>
   <hook_url>https://hooks.slack.com/services/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 </hook_url>
   <level>10</level>
   <alert_format>json</alert_format>
  </integration>


  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
  <logging>
    <log_format>plain</log_format>
  </logging>

  <remote>
    <connection>secure</connection>
S    <port>1514</port>
```

Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your message in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello,
World!"}'
https://hooks.slack.com/services/T09JQ1UKFEY/B09JL1GN62J/KMiRkvWfJsUOtlauiFj3Pcy
4
```

Copy

Webhook URL	Channel	Added By
https://hooks.slack.com/services/T0 Copy	#wazuh-alerts	Fazal Shaik Oct 3, 2025
https://hooks.slack.com/services/T0 Copy	#wazuh-alerts	Fazal Shaik Oct 3, 2025

Add New Webhook