IMPLEMENTATION OF WAZUH

Project Report

Submitted By:Esha Khadeeja C

Table of Contents

Executive Summary

This project report details the implementation of Wazuh, an open-source security monitoring platform, within an Ubuntu virtual machine using Docker containers. The objective was to enhance the security posture and perform comprehensive vulnerability assessments on two enrolled agents: a Windows host machine and a Kali Linux virtual machine.

The project began with the setup of Wazuh on an Ubuntu virtual machine, leveraging Docker containers for efficient deployment and management. Following the successful installation, two agents were enrolled within the Wazuh web interface. These agents, representing the Windows host machine and the Kali Linux virtual machine, were configured for continuous monitoring and vulnerability analysis.

The core of this project involved using Wazuh to monitor system activities and detect potential security threats in real-time. The Windows and Kali Linux agents were subjected to detailed vulnerability scans, identifying security weaknesses and potential exploits. The results of these scans provided critical insights into the security status of each system.

Key findings from the vulnerability assessments revealed several vulnerabilities of varying severity levels. These findings underscored the importance of regular security monitoring and timely remediation to protect against potential threats. Recommendations for mitigating identified vulnerabilities were provided, aiming to enhance the overall security of the monitored systems.

Introduction

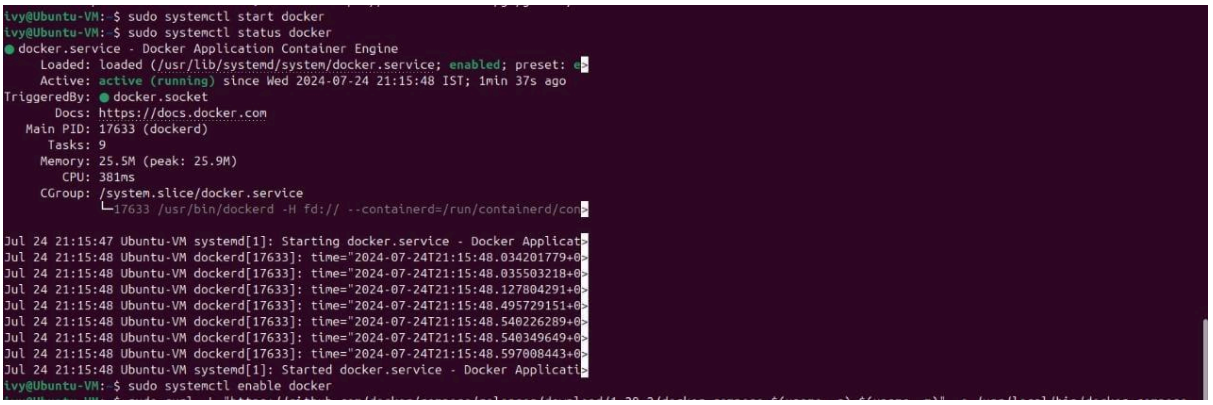
Security to information systems is indispensable amidst the changing digital scenery. Companies are resorting to powerful tools and techniques as a way of keeping their assets from potential threats or vulnerabilities. Wazuh is an open-source security monitoring platform that offers

comprehensive monitoring, threat detection, incident response, and compliance management.

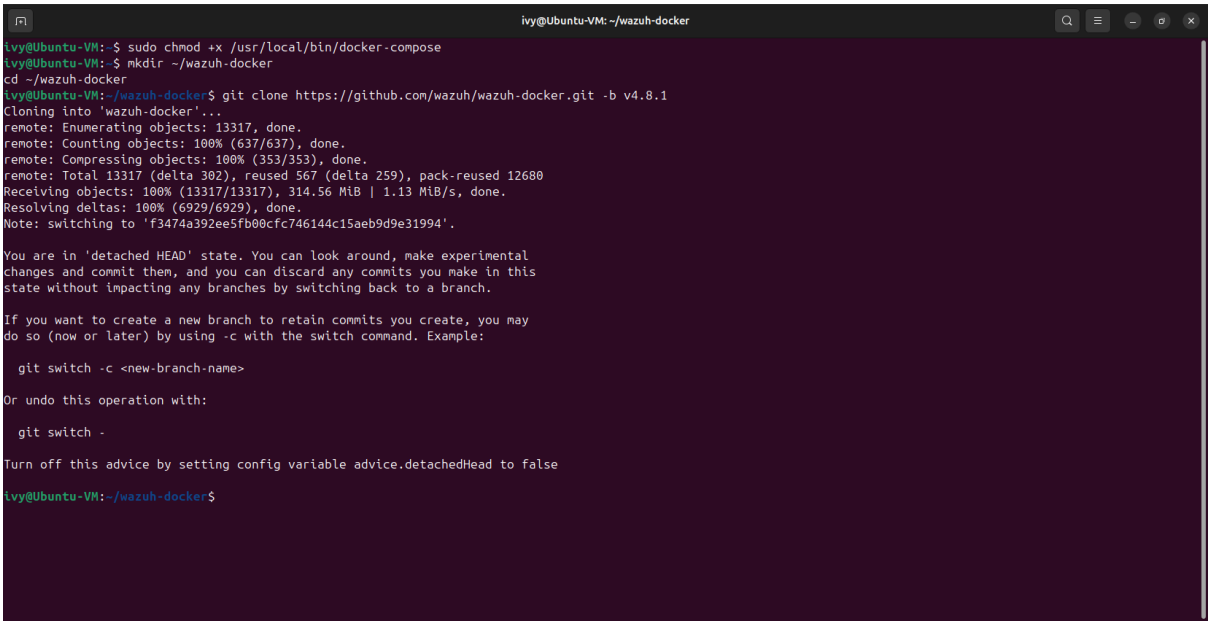
System Setup and Configuration

This section details the steps taken to implement Wazuh on an Ubuntu virtual machine using Docker container. And Enrolling agents.

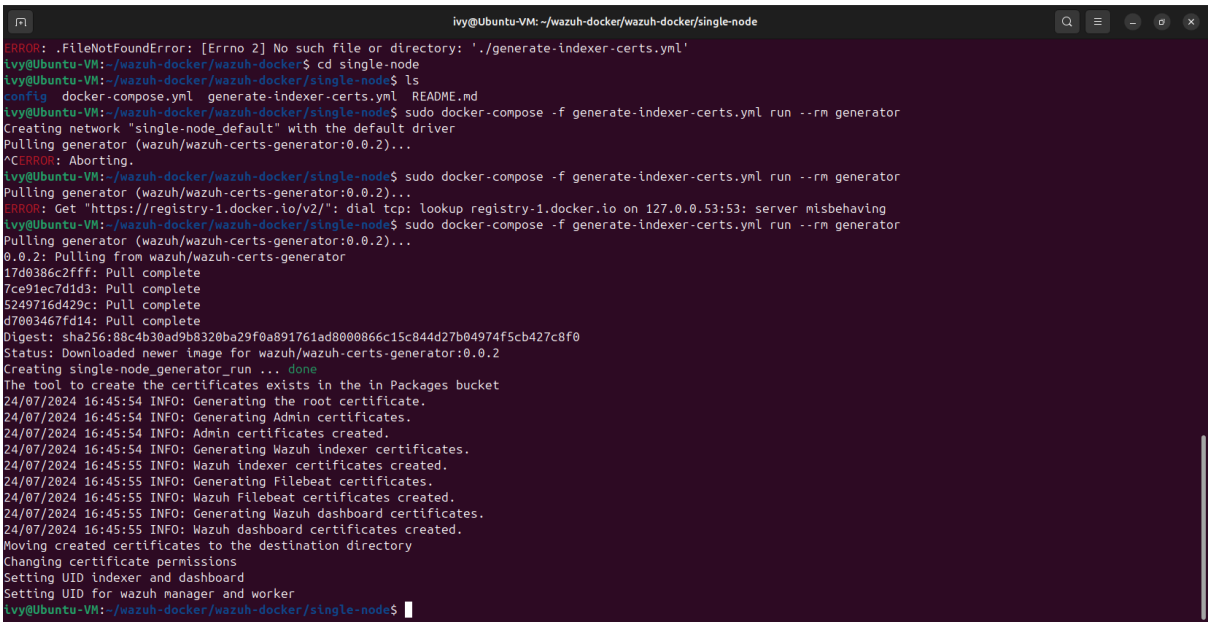
- Install Ubuntu Virtual machine
- Install Docker
    1. Update your package list:
        sudo apt-get update
    2. Install Docker
        sudo apt-get install docker.io
    3. Start and enable Docker:
        sudo systemctl start docker
        sudo systemctl enable docker



- Install Docker Compose
    1. Download the Docker Compose binary: *sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose*
- Download Wazuh Docker Repository
    1 . Clone the Wazuh Docker repository: *git clone https://github.com/wazuh/wazuh-docker.git -b v4.8.1*
    2. Change to wazuh-docker directory: *cd wazuh-docker*



- Generate self-signed certificates for each cluster node
    1. Change to single-node directory: *cd single-node*
    2. Get the desired certificates: *docker-compose -f generate-indexer-certs.yml run --rm generator*



- Deploy Wazuh

    1 .Run Docker Compose to start Wazuh:*sudo docker-compose up -d*

```
ivy@Ubuntu-VM:~/wazuh-docker/wazuh-docker/single-node$ sudo docker-compose up -d
Creating volume "single-node_wazuh_api_configuration" with default driver
Creating volume "single-node_wazuh_etc" with default driver
Creating volume "single-node_wazuh_logs" with default driver
Creating volume "single-node_wazuh_queue" with default driver
Creating volume "single-node_wazuh_var_multigroups" with default driver
Creating volume "single-node_wazuh_integrations" with default driver
Creating volume "single-node_wazuh_active_response" with default driver
Creating volume "single-node_wazuh_agentless" with default driver
Creating volume "single-node_wazuh_wodles" with default driver
Creating volume "single-node_filebeat_etc" with default driver
Creating volume "single-node_filebeat_var" with default driver
Creating volume "single-node_wazuh-indexer-data" with default driver
Creating volume "single-node_wazuh-dashboard-config" with default driver
Creating volume "single-node_wazuh-dashboard-custom" with default driver
Pulling wazuh.manager (wazuh/wazuh-manager:4.8.1)...
```

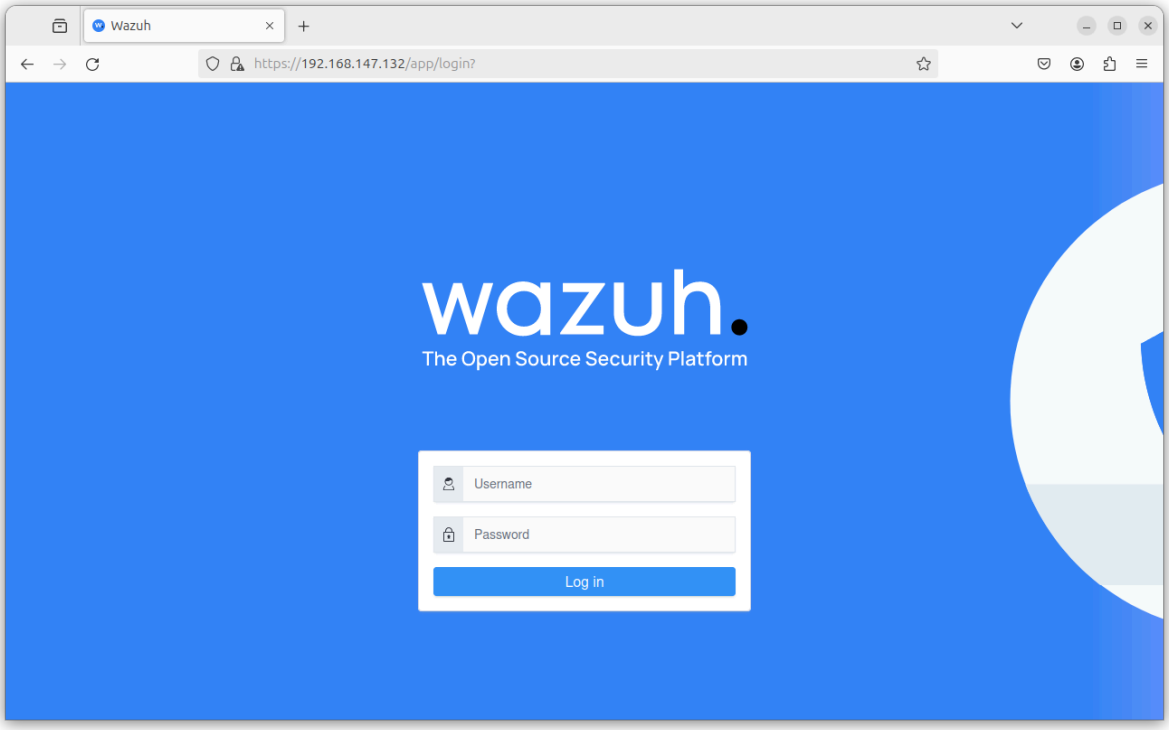
```
6cb06adbaaac: Pull complete
f3408027b26a: Pull complete
43b2c6c42c73: Pull complete
6da242b399ab: Pull complete
23795cd84012: Pull complete
e484a6a0f881: Pull complete
4cef5085f033: Pull complete
9748b6970731: Pull complete
955315937cf5: Pull complete
28c81a5731c2: Pull complete
d96ae6a63283: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:dfdee4d4c4219f3fd68a9c1015e817556b0a9d8f67b30520d44dfb2619d355cc
Status: Downloaded newer image for wazuh/wazuh-dashboard:4.8.1
Creating single-node_wazuh.manager_1 ... done
Creating single-node_wazuh.indexer_1 ... done
Creating single-node_wazuh.dashboard_1 ... done
```

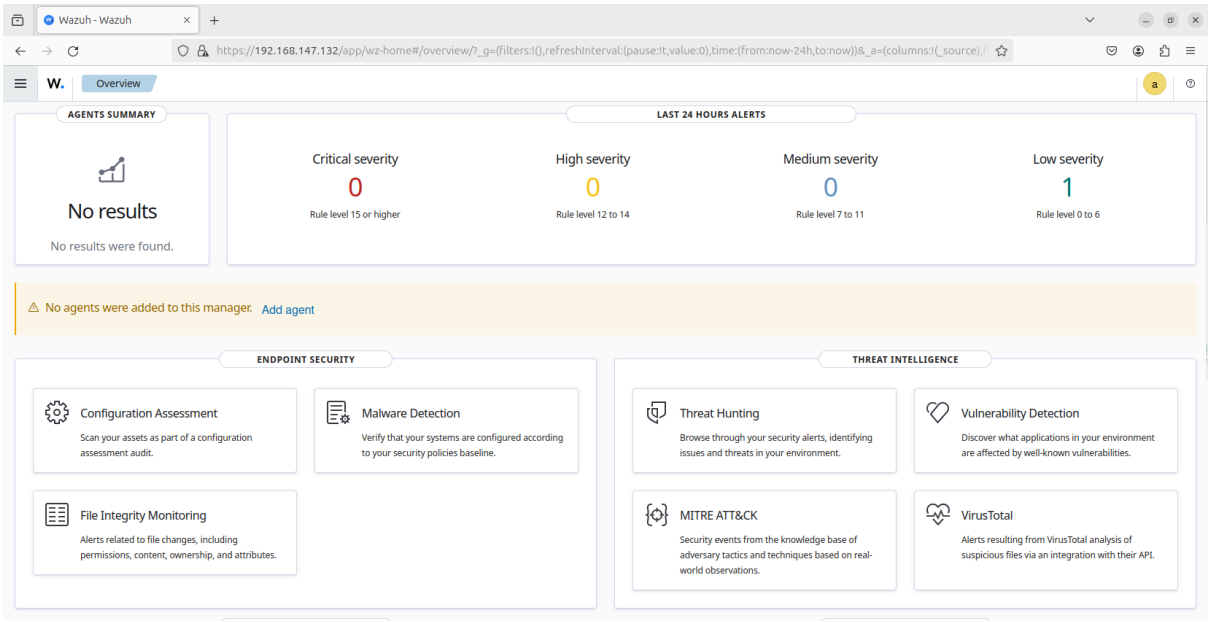
2 .Verify the containers are running:

sudo docker ps -a

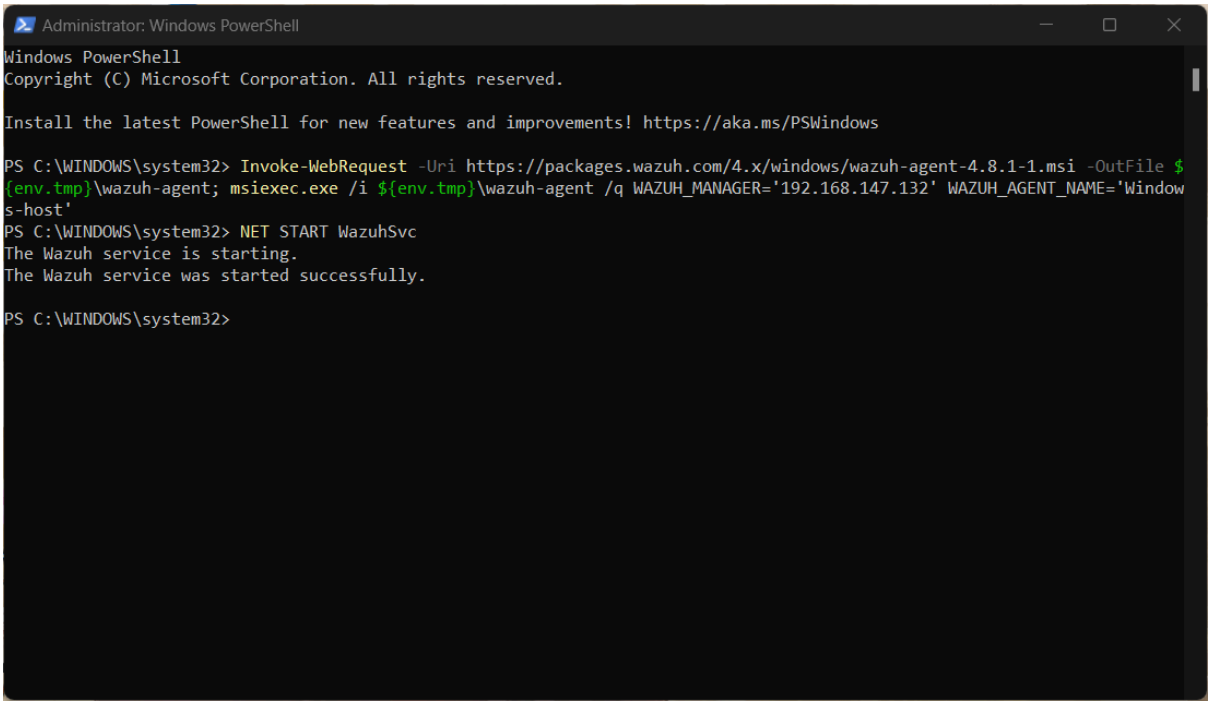
```
ivy@Ubuntu-VM: ~/wazuh-docker/wazuh-docker/single-node
ivy@Ubuntu-VM:~/wazuh-docker/wazuh-docker/single-node$ docker ps -a
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/json?all=1": dial unix /var/run/docker.sock: connect: permission denied
ivy@Ubuntu-VM:~/wazuh-docker/wazuh-docker/single-node$ sudo docker ps -a
[sudo] password for ivy:
CONTAINER ID   IMAGE                       COMMAND                  CREATED          STATUS          PORTS
                                                                                                    NAMES
3d77f531e6f7   wazuh/wazuh-dashboard:4.8.1   "/entrypoint.sh"         42 minutes ago   Up 42 minutes   443/tcp, 0.0.0.0:443->5601/tcp, :::443->5601/tcp
                                                                                                    single-node_wazuh.dashboard_1
141e734bd877   wazuh/wazuh-manager:4.8.1     "/init"                  42 minutes ago   Up 42 minutes   0.0.0.0:1514-1515->1514-1515/tcp, :::1514-1515->1514-1515/tcp, 0.0.0.0:514->514/udp, :::514->514/udp, 0.0.0.0:55000->55000/tcp, :::55000->55000/tcp, 1516/tcp   single-node_wazuh.manager_1
b90e5e4b0875   wazuh/wazuh-indexer:4.8.1     "/entrypoint.sh open…"   42 minutes ago   Up 42 minutes   0.0.0.0:9200->9200/tcp, :::9200->9200/tcp
                                                                                                    single-node_wazuh.indexer_1
ivy@Ubuntu-VM:~/wazuh-docker/wazuh-docker/single-node$
```

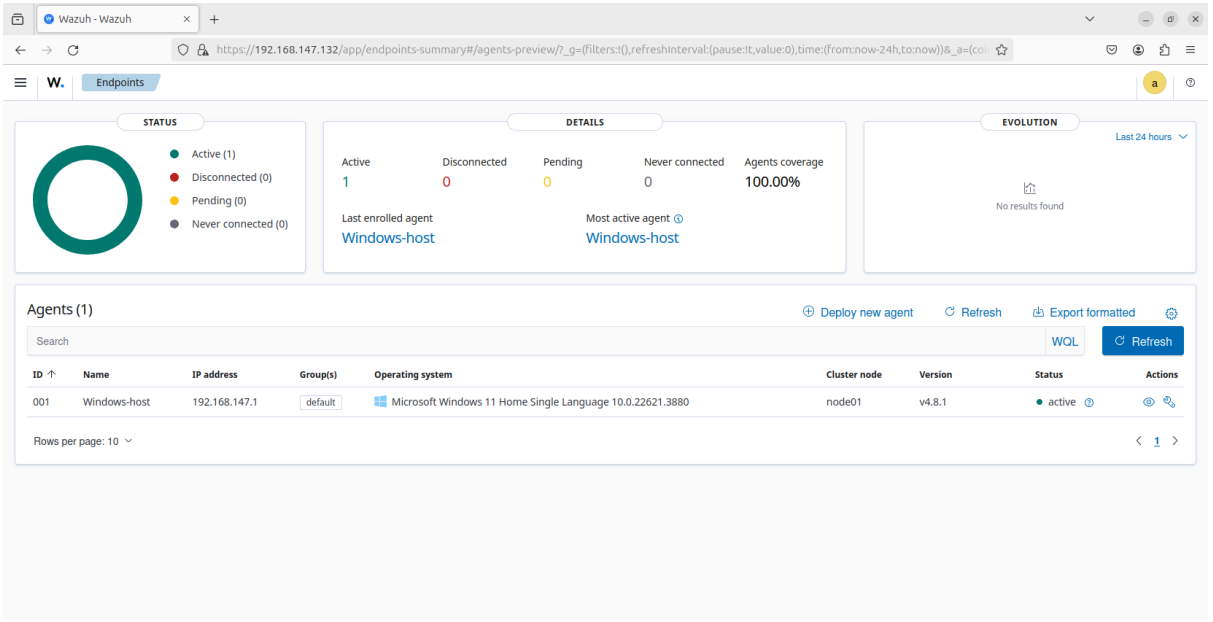
- Access the Wazuh Web Interface
    1. Open web browser and navigate to the interface URL:https://<ubuntu server ip>
    2. Login with the credentials





- Enroll Agents
    1. Add Windows 11(Host Machine) as an agent
    2. Give information required such as os of the agent, server ip address, name for the agent
    3. Run the code given in the agent machine's terminal

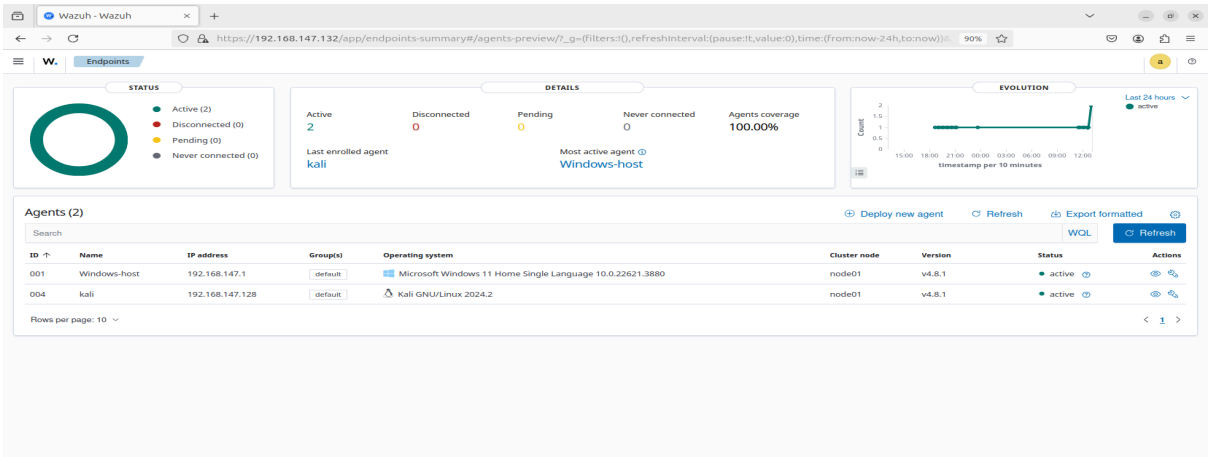




4. Add Kali Linux Virtual Machine as an Agent
5. Run the code in kali linux bash


kali@kali: ~
File Actions Edit View Help
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh- agent_4.8.1-1_amd64.deb && sudo WAZUH_MANAGER='192.168.147.132' WAZUH_AGENT_NAME='Kali_Linux_Attacker' dpkg -i ./wazuh-agent_4.8.
64.deb
--2024-07-29 10:40:19-- https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-
Resolving packages.wazuh.com (packages.wazuh.com)... 13.33.146.9, 13.33.146.56, 13.33.146.28, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|13.33.146.9|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2024-07-29 10:40:21 ERROR 404: Not Found.
--2024-07-29 10:40:21-- http://agent_4.8.1-1_amd64.deb/
Resolving agent_4.8.1-1_amd64.deb (agent_4.8.1-1_amd64.deb)... failed: Name or service not known.
wget: unable to resolve host address 'agent_4.8.1-1_amd64.deb'
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.8.1-1_amd64.deb && sudo WAZUH_MANAGER='192.168.147.132' WAZUH_AGENT_NAME='Kali_Linux_Attacker' dpkg -i ./wazuh-agent_4.8.1
4.deb
--2024-07-29 10:40:33-- https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.8.1-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 13.33.146.85, 13.33.146.28, 13.33.146.56, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|13.33.146.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10270680 (9.8M) [binary/octet-stream]
Saving to: 'wazuh-agent_4.8.1-1_amd64.deb.3'
wazuh-agent_4.8.1-1_amd64.deb.3 100%[ ] 9.79M 15.1MB/s in 0.
2024-07-29 10:40:33 (15.1 MB/s) - 'wazuh-agent_4.8.1-1_amd64.deb.3' saved [10270680/10270680]
[sudo] password for kali:
(Reading database ... 394193 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.8.1-1_amd64.deb ...
Unpacking wazuh-agent (4.8.1-1) over (4.8.1-1) ...
Setting up wazuh-agent (4.8.1-1) ...

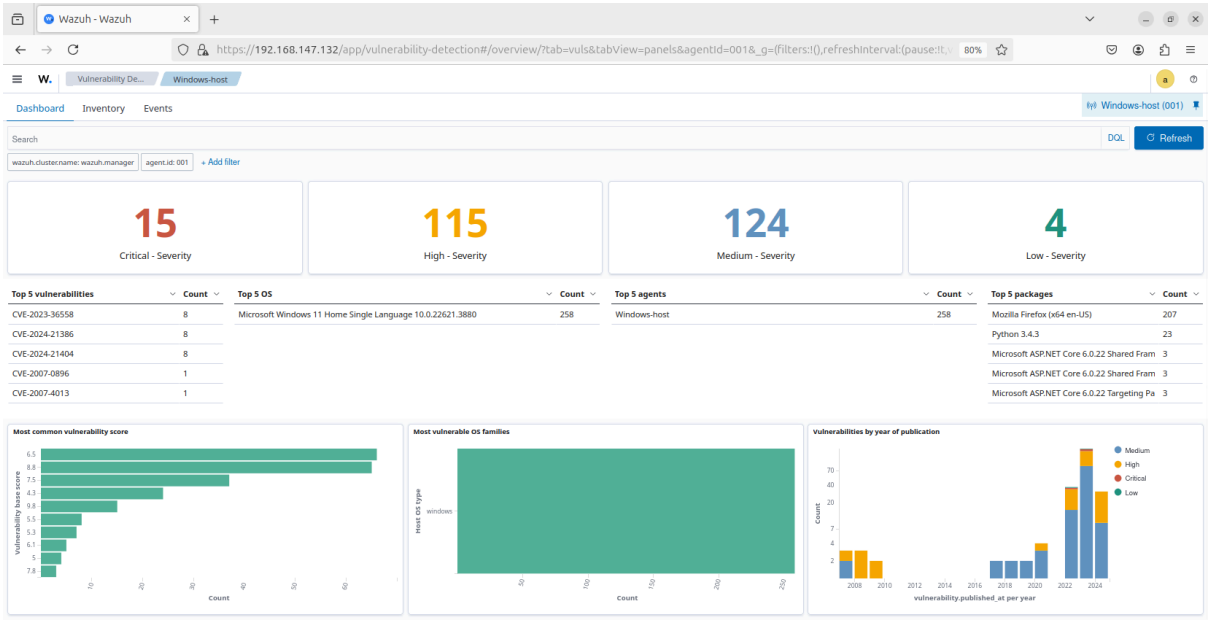

kali@kali: ~
File Actions Edit View Help
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /usr/lib/systemd/system/wazuh-agent.service.
sudo systemctl start wazuh-agent


Wazuh - Wazuh
https://192.168.147.132/app/endpoints-summary#/agents-preview/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-24h,to:now))
Endpoints
STATUS
Active (2)
Disconnected (0)
Pending (0)
Never connected (0)
DETAILS
Active 2
Disconnected 0
Pending 0
Never connected 0
Agents coverage 100.00%
Last enrolled agent kali
Most active agent Windows-host
EVOLUTION
Last 24 hours
timestamp per 10 minutes
Agents (2)
Deploy new agent
Refresh
Export formatted
Search
WQL
Refresh
ID Name IP address Group(s) Operating system Cluster node Version Status Actions
001 Windows-host 192.168.147.1 default Microsoft Windows 11 Home Single Language 10.0.22621.3880 node01 v4.8.1 active
004 kali 192.168.147.128 default Kali GNU/Linux 2024.2 node01 v4.8.1 active
Rows per page: 10

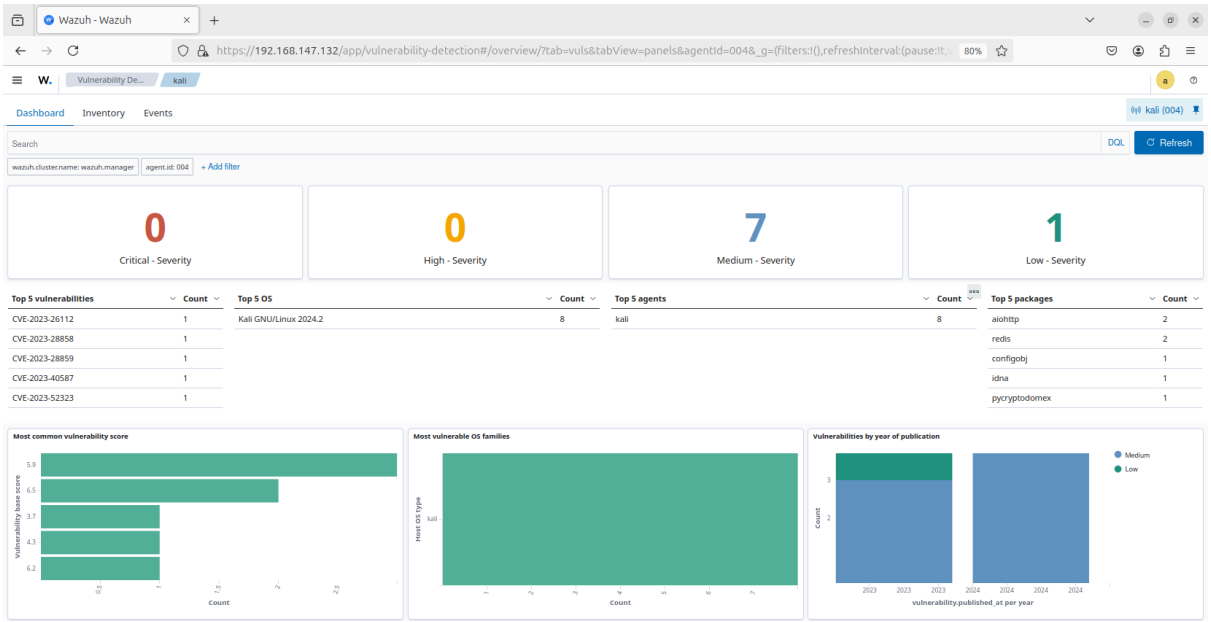
Vulnerability Scanning

This section details the process of conducting vulnerability scans using Wazuh on the enrolled agents: a Windows host machine and a Kali Linux virtual machine.

- Initiate Vulnerability Scan on Windows Machine



- Initiate Vulnerability Scan on Kali Linux



Summary of Findings:

1. **Kali Linux Agent**:
    - **Severity Levels**:
        - Critical: 0
        - High: 0
        - Medium: 7
        - Low: 1
    - **Top 5 Vulnerabilities**:
        - CVE-2023-26112
        - CVE-2023-28858
        - CVE-2023-28859
        - CVE-2023-40587
        - CVE-2023-52323
    - **Top 5 Packages**:
        - aiohttp: 2
        - redis: 2
        - configobj: 1
        - idna: 1
        - pycryptodomex: 1
    - The most common vulnerability base scores ranged from 3.7 to 6.5.
2. **Windows Host Machine**:
    - **Severity Levels**:
        - Critical: 15
        - High: 115
        - Medium: 124
        - Low: 4
    - **Top 5 Vulnerabilities**:
        - CVE-2023-36558
        - CVE-2024-21386
        - CVE-2024-21404
        - CVE-2007-0896
        - CVE-2007-4013
    - **Top 5 Packages**:

        - Mozilla Firefox (x64 en-US): 207
        - Python 3.4.3: 23
        - Microsoft ASP.NET Core 6.0.22 Shared Framework: 3
    - The most common vulnerability base scores ranged from 5.3 to 8.8.

Analysis:

- **Kali Linux Agent**: The vulnerability scan revealed a total of eight vulnerabilities, with none categorized as critical or high. Most of the vulnerabilities fall under the medium severity level, with a few low-severity issues. The most vulnerable packages are related to web and cryptographic libraries, which should be promptly updated to mitigate potential risks.
- **Windows Host Machine**: The Windows host machine showed a significantly higher number of vulnerabilities, with 15 critical, 115 high, and 124 medium severity issues. This indicates a higher risk profile compared to the Kali Linux agent. The critical vulnerabilities, especially those affecting widely used packages like Mozilla Firefox and Microsoft ASP.NET Core, need immediate attention to prevent potential exploitation.

Recommendations:

1. **For Kali Linux:**
    - Regularly update the system and installed packages to the latest versions.
    - Focus on upgrading or patching the identified vulnerable packages (aiohttp, redis, configobj, etc.).
    - Implement continuous monitoring to promptly detect and respond to new vulnerabilities.

2. **For Windows Host Machine**:
    - Immediate patching of critical and high-severity vulnerabilities is crucial.
    - Ensure that all software, particularly those identified as vulnerable (Mozilla Firefox, Python, ASP.NET Core), is kept up to date.
    - Strengthen the overall security posture by implementing additional security measures such as firewall configurations, anti-malware solutions, and system hardening practices.

Conclusion:

This project successfully demonstrated the deployment of Wazuh on an Ubuntu virtual machine using Docker containers and the effective use of Wazuh for vulnerability scanning on a Windows host machine and a Kali Linux virtual machine. The key achievements include:

1. **Deployment of Wazuh**: Wazuh was deployed using Docker containers on an Ubuntu virtual machine, providing a scalable security monitoring solution.

2. **Agent Enrollment**: The Windows host machine and Kali Linux virtual machine were successfully enrolled and configured as Wazuh agents.

3. **Vulnerability Scanning**: Effective vulnerability scans were conducted, identifying several security vulnerabilities and providing actionable mitigation steps.

The project highlighted Wazuh's capability in enhancing system security through continuous monitoring and vulnerability assessment. The outcomes provide a foundation for ongoing security improvements.