

# Achieving Cyber Resilience

**Achieving Cyber Resilience:** Building an Effective SIEM with Open-Source Tools

## **Contexts:**

<b>1</b>	<b>Introduction</b>	
1.1	Prerequisites .....	4
1.2	Features of Wazuh .....	5
<b>2</b>	<b>Wazuh</b>	
2.1	Installation of Wazuh .....	7
2.2	Deploying Wazuh .....	9
2.3	Network configuration .....	11
2.4	Wazuh-manager configuration .....	15
2.5	Wazuh-indexer configuration .....	16
<b>3</b>	<b>Wazuh Server</b>	
3.1	Deploying agentes .....	17
3.2	Wazuh-agent configuration .....	18
<b>4</b>	<b>Suricata</b>	
4.1	Installation of suricata .....	20
4.2	Suricata rules .....	22
4.3	Suricata configuration .....	23
<b>5</b>	<b>Ubuntu with Json</b>	
5.1	Installation of Json .....	30
5.2	Configuration of Json .....	30
<b>6</b>	<b>Zeek Integration with ELK stack</b>	
6.1	Zeek repositories .....	32
6.2	Installation of Zeek .....	33
6.3	Configuration of Zeek .....	35
6.4	Zeek logs updatation .....	36
6.5	Filebeat configuration .....	40
<b>7</b>	<b>Elasticsearch</b>	
7.1	Introduction of elasticsearch .....	43
7.2	Installation of elasticsearch .....	45
7.3	Configuration of elasticsearch .....	46
<b>8</b>	<b>Wazuh cluster</b>	
8.1	Importing repositories .....	48
8.2	Adding repositories .....	48

## **9 Kibana**

9.1	Prerequisites and repositories .....	49
9.2	Installation of kibana .....	50
9.3	Directory .....	51
9.4	Configuration of kibana. ....	52
9.5	Adding rules set.....	52

## **Abstract:**

System monitoring is a critical component of cybersecurity, as it allows organizations to detect and respond to threats in real time. By continuously monitoring system activity, organizations can identify anomalies that may indicate a security breach, and take appropriate action to mitigate the threat. System monitoring can be performed manually or via automated tools, and should be tailored to the specific needs of the organization.

## **Introduction:**

In the modern technology cybersecurity plays a major role. Monitoring and preventing the cyber-attacks are the important role in the cybersecurity. So, in this part we are going to show how we can monitor the end users (Clients) with the help of the open-source platforms.

## **System Requirements:**

- Ram: 16-32 GB RAM
- Processor: i7-i9 (Intel (OR) AMD RYZEN 5)
- Storage: Min 512GB
- Operating System: Windows, MacOS

## **Software's Required:**

- Wazuh
- Ubuntu 22.04
- VMware workstation
- Zeek
- Suricata

## **Introduction to wazuh:**

Wazuh is a free and open-source platform used for threat prevention, detection, and response. It is capable of protecting workloads across on-premises, virtualized, containerized, and cloud based environments. Wazuh solution consists of an endpoint security agent, deployed to the monitored systems, and a management server, which collects and analyses data gathered by the agents. Besides, Wazuh has been fully integrated with the Elastic Stack, providing a search engine and data visualization tool that allows users to navigate through their security alerts.

In the context of blue team operations, wazuh is a **SIEM (Security Information Event Management)** system that is used to collect, analyse, aggregate, index and analyse security related data consequently allowing you to detect intrusions, attacks, vulnerabilities, and malicious activity. We will be using Wazuh to monitor security events and identify vulnerabilities on agents.

## **Use of wazuh:**

### **Wazuh's Features:**

- Security Analytics

Wazuh is used to collect, aggregate, index and analyse security data, helping organizations detect intrusions, threats and behavioural anomalies. As cyber threats are becoming more sophisticated, real-time monitoring and security analysis are needed for fast threat detection and remediation

- Intrusion Detection

Wazuh agents scan the monitored systems looking for malware, rootkits and suspicious anomalies. They can detect hidden files, cloaked processes or unregistered network listeners, as well as inconsistencies in system call responses. In addition to agent capabilities, the server component uses a signature-based approach to intrusion detection, using its regular expression engine to analyse collected log data and look for indicators of compromise.

- Log Data Analysis

Wazuh agents read operating system and application logs, and securely forward them to a central manager for rule-based analysis and storage. The Wazuh rules help make you aware of application or system errors, misconfigurations attempted and/or successful malicious activities, policy violations and a variety of other security and operational issues

- File Integrity Monitoring

Wazuh monitors the file system, identifying changes in content, permissions, ownership, and attributes of files that you need to keep an eye on. In addition, it natively identifies users and applications used to create or modify files. File integrity monitoring capabilities can be used in combination with threat intelligence to identify threats or compromised hosts. In addition, several regulatory compliance standards, such as PCI (Payment Card Industry), DSS (Data Security Standard), require it.

- **Vulnerability Detection**

Wazuh agents pull software inventory data and send this information to the server, where it is correlated with continuously updated CVE (Common Vulnerabilities and Exposure) databases, to identify well-known vulnerable software. Automated vulnerability assessment helps you find the weak spots in your critical assets and take corrective action before attackers exploit them to sabotage your business or steal confidential data.

- **Configuration Assessment**

Wazuh monitors system and application configuration settings to ensure they are compliant with your security policies, standards and/or hardening guides. Agents perform periodic scans to detect applications that are known to be vulnerable, unpatched, or insecurely configured. Additionally, configuration checks can be customized, tailoring them to properly align with your organization. Alerts include recommendations for better configuration, references, and mapping with regulatory compliance.

- **Incident Response**

Wazuh provides out-of-the-box active responses to perform various countermeasures to address active threats, such as blocking access to a system from the threat source when certain criteria are met. In addition, Wazuh can be used to remotely run commands or system queries, identifying indicators of compromise (IOCs) and helping perform other live forensics or incident response tasks.

- **Cloud Security**

Wazuh helps monitoring cloud infrastructure at an API(Application programming interface) level, using integration modules that can pull security data from well-known cloud providers, such as Amazon AWS, Azure or Google Cloud. In addition, Wazuh provides rules to assess the configuration of your cloud environment, easily spotting weaknesses. In addition, Wazuh lightweight and multi-platform agents are commonly used to monitor cloud environments at the instance level.

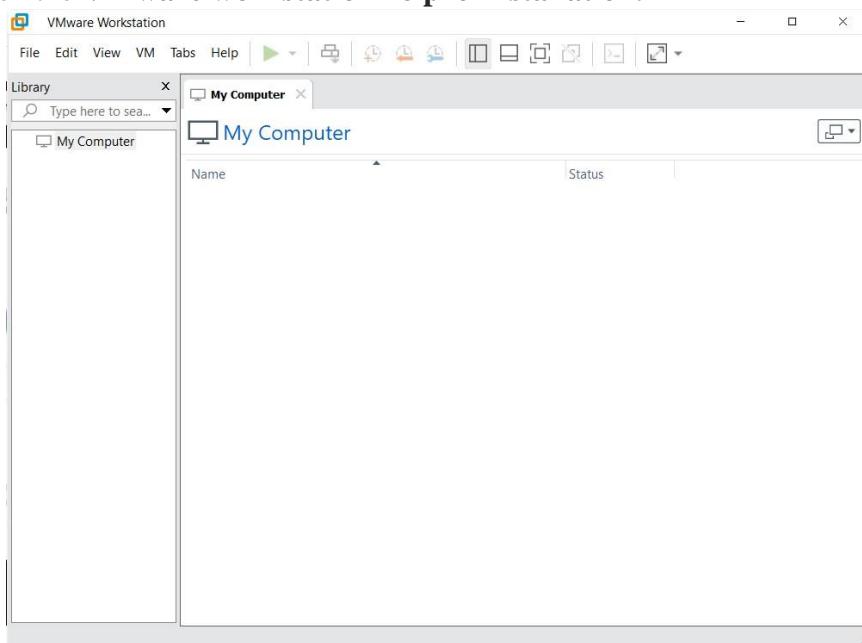
- **Installation and configuring the wazuh in VMWare:**

**Select your own Virtual Machine:** VMware workstation 16 pro.

**Link:** <https://www.vmware.com/in/products/workstation-pro/workstation-proevaluation.html>

**After installation of VMware workstation 16 pro Installation.**

- Open the VMware workstation 16 pro Installation.

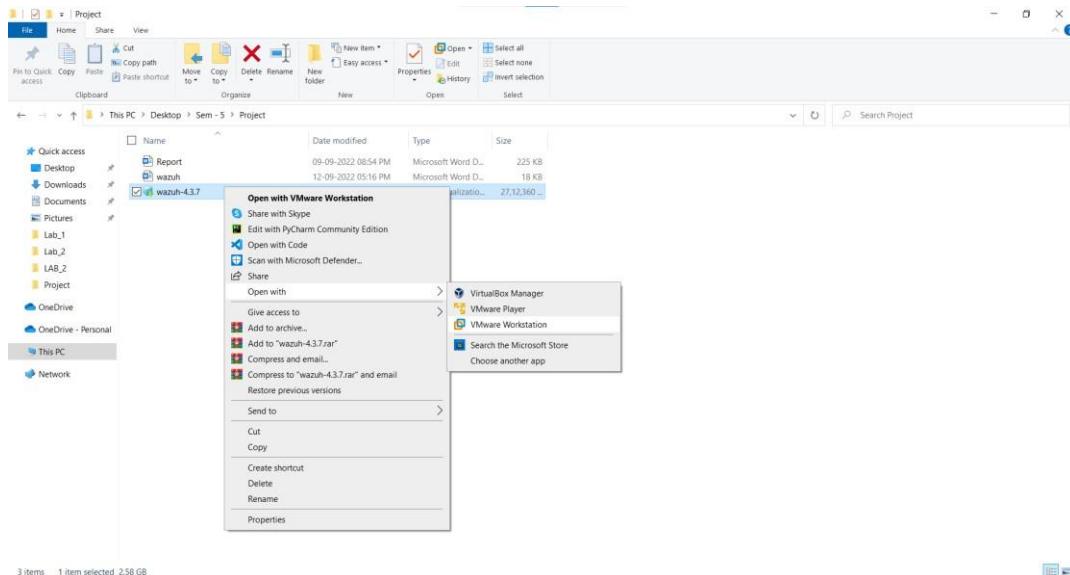


**Download the wazuh-4.3.7.ova**

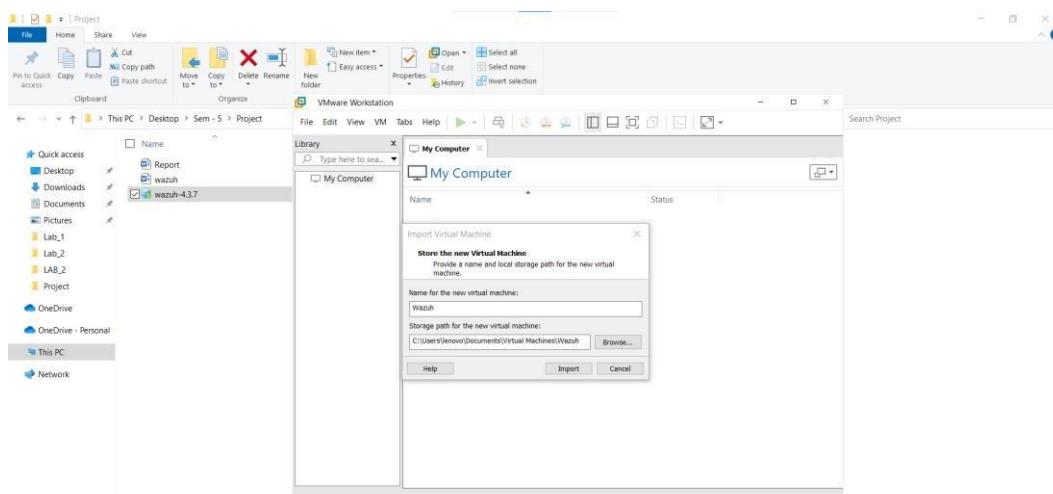
**Link:** <https://documentation.wazuh.com/current/deployment-options/virtualmachine/virtual-machine.html>

A screenshot of a web browser displaying the Wazuh documentation page. The URL is 'https://documentation.wazuh.com/current/deployment-options/virtualmachine/virtual-machine.html'. The page title is 'Virtual Machine (OVA)'. The left sidebar has a 'Virtual Machine (OVA)' section highlighted. The main content area starts with a heading 'Virtual Machine (OVA)' and a paragraph about OVA files. It then says 'Download the virtual appliance (OVA)', which is highlighted with a red box. Below this, a list of components is provided: CentOS 7, Wazuh manager 4.3.7, Wazuh indexer 4.3.7, Filebeat-OSS 7.10.2, and Wazuh dashboard 4.3.7. At the bottom of the page, there is a 'Hardware requirements' section.

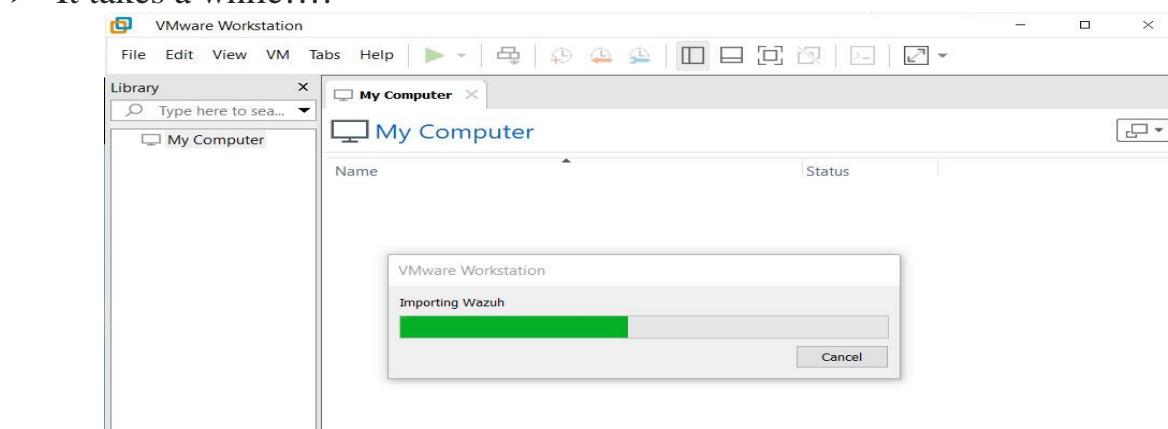
- Open the Wazuh-4.3.7.ova file with VMware workstation 16 pro.



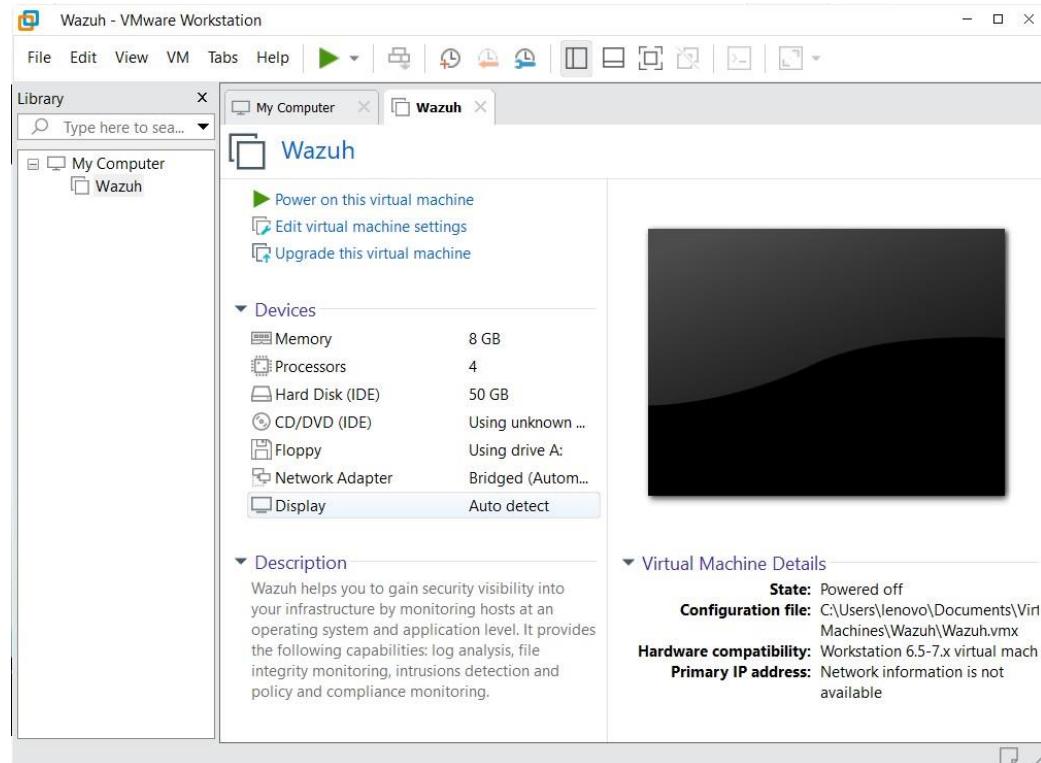
**Step 1:** Enter the Virtual machine name and leave Storage path for default: Select Import.



- It takes a while!!!!



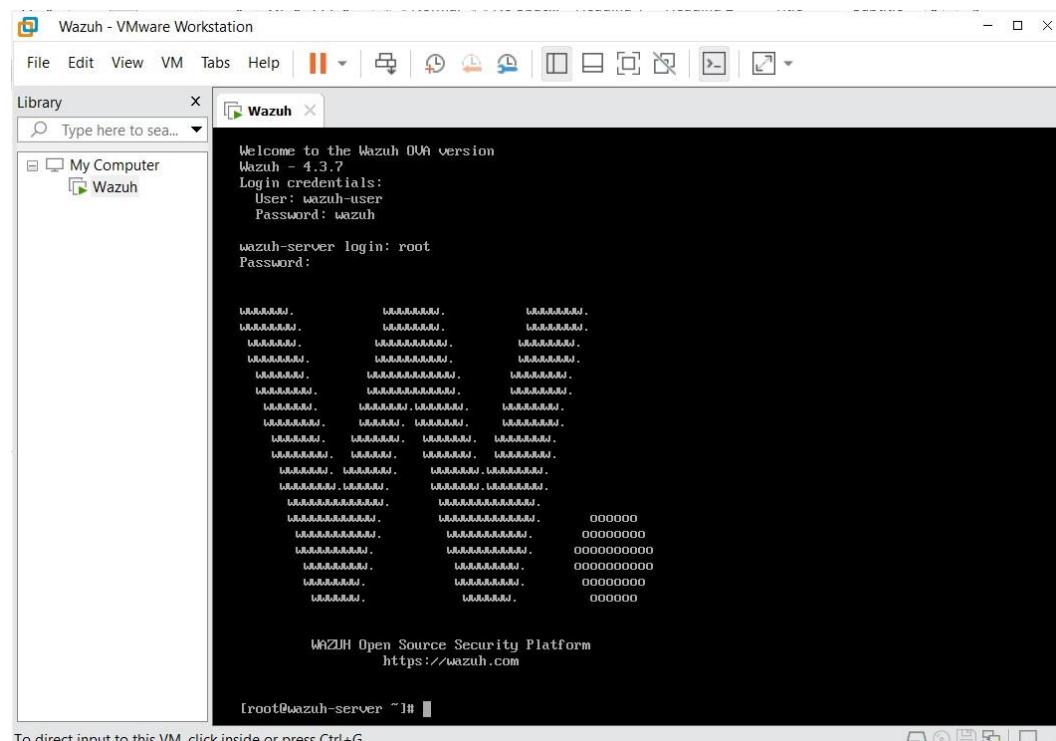
## Step 2: After adding the Wazuh to VMware: Power on this virtual machine



## Step 3: Enter the wazuh-server login details:

**Username:** root

**Password:** wazuh ( By default the wazuh- server is having the username and the password)



**Step 5:** Run the command “Ip a” to check the weather server is producing the Ip. If the server is not producing any Ip change the Network setting in the Networks)

```
Wazuh - VMware Workstation
File Edit View VM Tabs Help | 
Library x
Type here to search...
My Computer
Wazuh

Wazuh

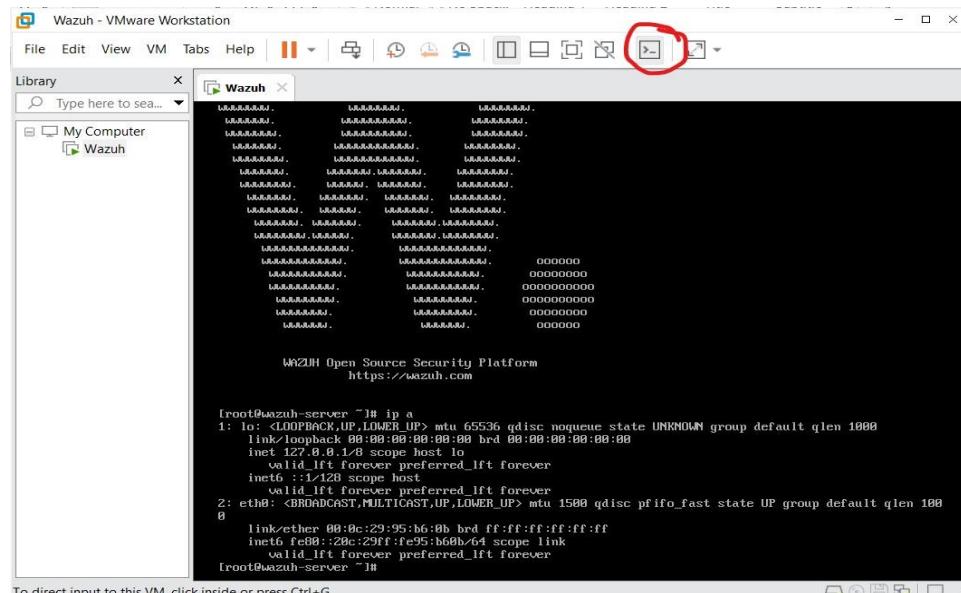
WAZUH Open Source Security Platform
https://wazuh.com

root@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 brd 127.255.255.255 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 brd 0000:0000:0000:0000:0000:0000:0000:0001 scope link
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:b5:b6:8b brd ff:ff:ff:ff:ff:ff
        inet6 fe80::20c:29ff:fe95:b68b/64 brd fe80::ff:ff:ff:ff:ff:ff scope link
            valid_lft forever preferred_lft forever
root@wazuh-server ~#
```

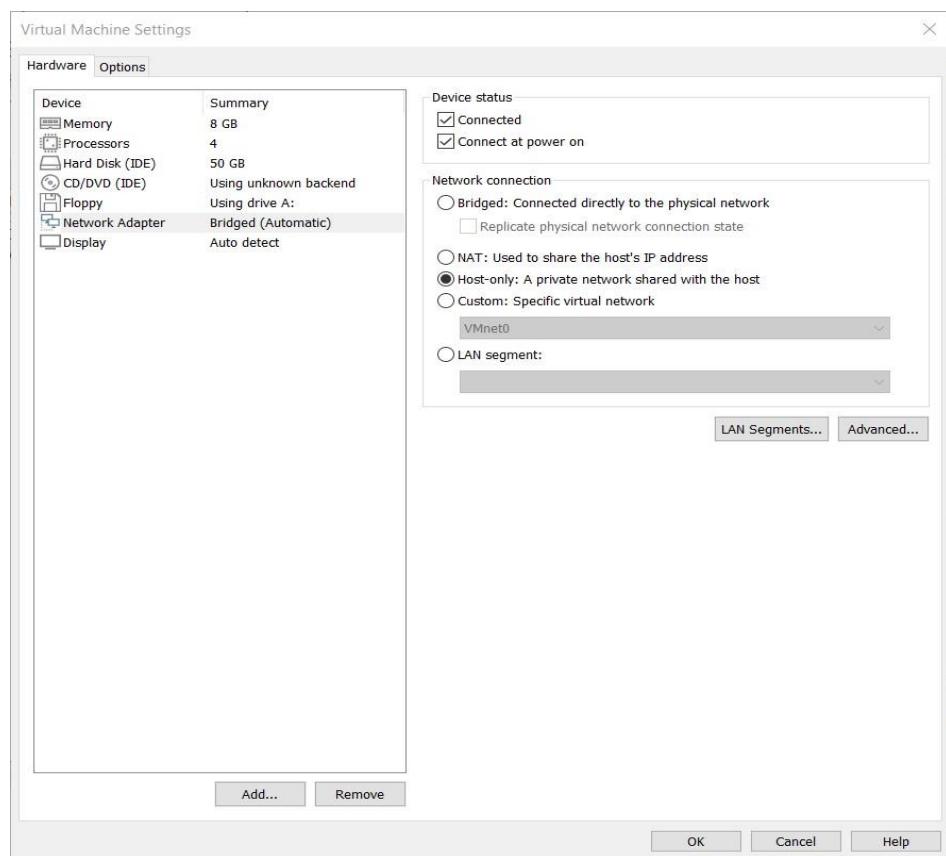
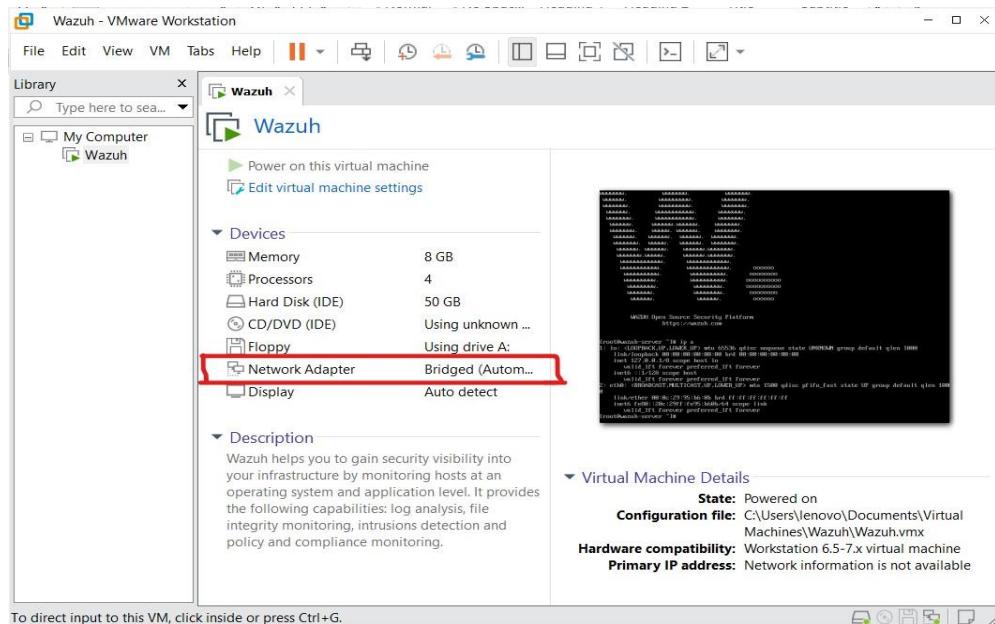
**Step 6:** Please ignore above steps if you got Ip(Only roman number): Continue there after!

In above server there is no Ip available because the server was running in vm which it was having different network. So, we need to change the network to the host-only network.

- Select the Show or hide the console view

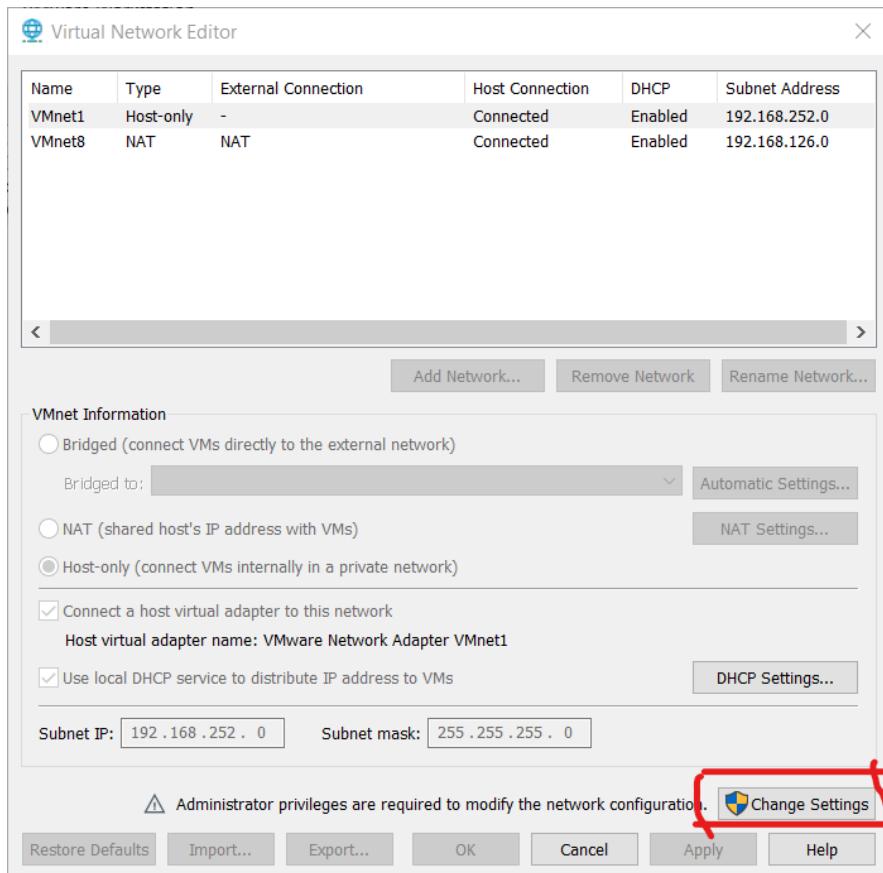
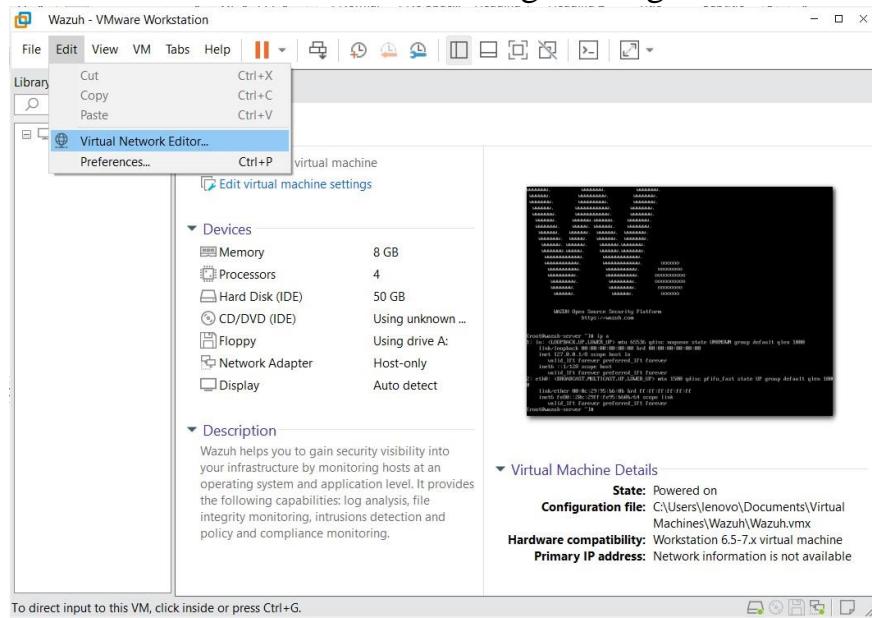


➤ Select Network Adapter to change the Network: **Select the Host-only Network**

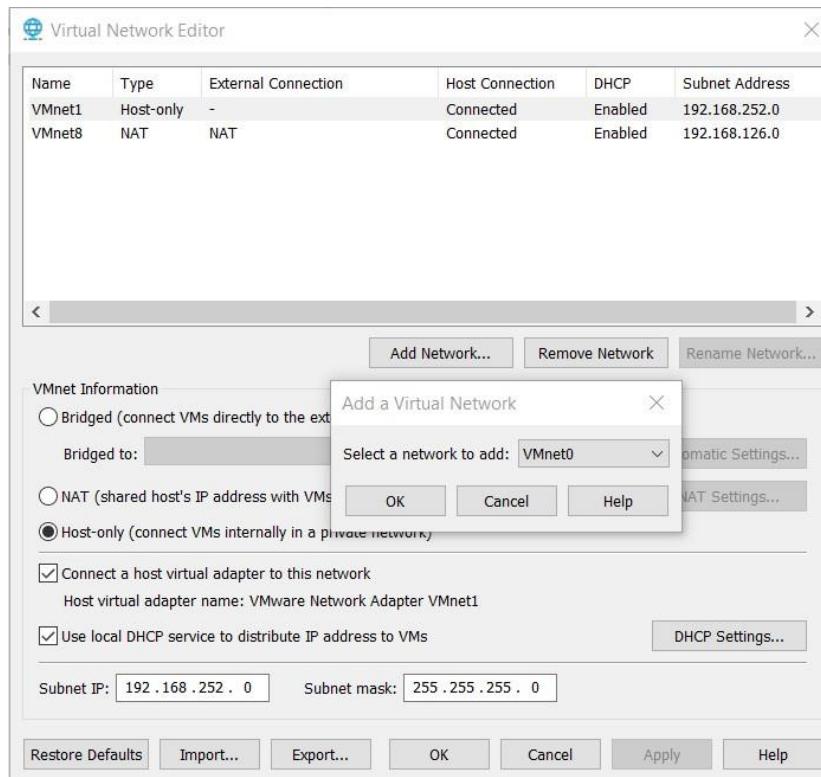


- Turn off the wazuh-server and edit the Network settings

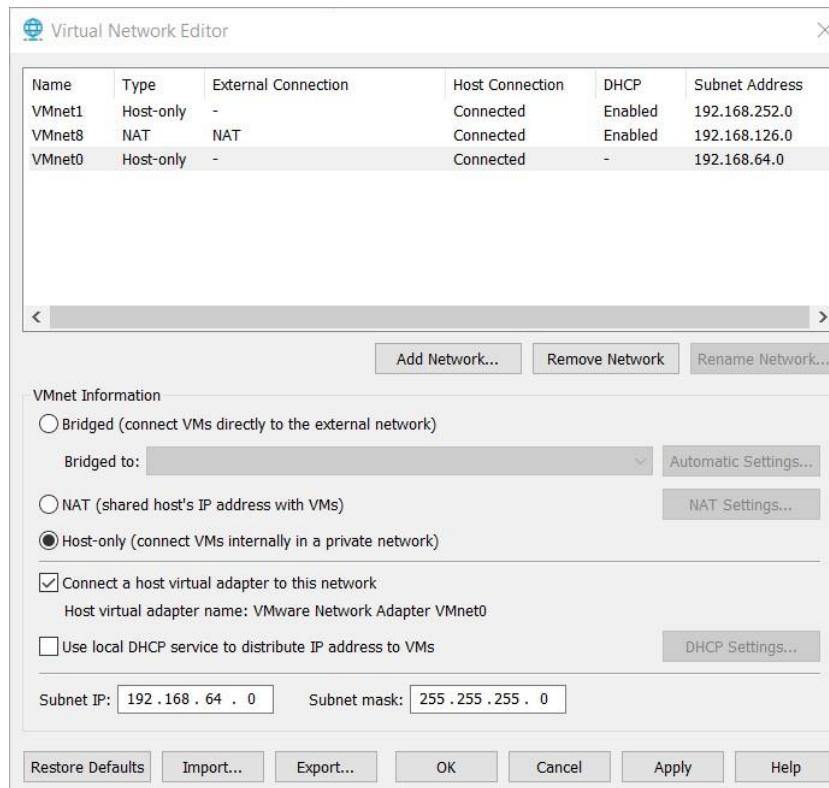
## Edit<Virtual Network Editor<change settings



- Add network<VMnet0<ok!



- Makes changes as the above settings and save the configuration.



- Now Reboot the wazuh and run the command to get the ip

**Step 7:** Here is your Ip to get the web dashboard: open your browser and enter <https://<ip>> Example: <https://192.168.0.123/>

- Wazuh-dashboard we will be opened. If you are facing any issues, please follow the following steps.
- If you are not able to open the dashboard, then use the above command to activate it #**systemctl status wazuh-manager**

```
[root@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
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:d7:33:d5 brd ff:ff:ff:ff:ff:ff
        inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic eth0
            valid_lft 592sec preferred_lft 592sec
        inet6 fe80::a00:27ff:fed7:33d5/64 scope link
            valid_lft forever preferred_lft forever
[root@wazuh-server ~]# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
  Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: disabled)
  Active: activating (start) since Mon 2022-09-12 23:09:54 UTC; 41s ago
    Control: 752 (wazuh-control)
   CGroup: /system.slice/wazuh-manager.service
           └─ 752 /bin/sh /var/ossec/bin/wazuh-control start
             ├─ 1279 /bin/sh /var/ossec/bin/wazuh-apid
             ├─ 1285 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py

Sep 12 23:10:30 wazuh-server env[752]: Starting Wazuh v4.3.7...
Sep 12 23:10:34 wazuh-server env[752]: wazuh-apid: Process 1308 not used by Wazuh, removing...
[root@wazuh-server ~]#
```

- Run the command to start the wazuh-manager “ # **systemctl start wazuh-manager** ”

```
[root@wazuh-server ~]# systemctl start wazuh-manager
[root@wazuh-server ~]# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
  Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: disabled)
  Active: active (running) since Mon 2022-09-12 23:10:58 UTC; 5h 28min left
    Process: 752 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/wazuh-manager.service
           ├─ 1301 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─ 1345 /var/ossec/bin/wazuh-authd
           ├─ 1361 /var/ossec/bin/wazuh-db
           ├─ 1391 /var/ossec/bin/wazuh-execd
           ├─ 1407 /var/ossec/bin/wazuh-analysisd
           ├─ 1429 /var/ossec/bin/wazuh-syscheckd
           ├─ 1448 /var/ossec/bin/wazuh-remoted
           ├─ 1455 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─ 1458 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─ 1520 /var/ossec/bin/wazuh-logcollector
           ├─ 1534 /var/ossec/bin/wazuh-monitorfd
           ├─ 1548 /var/ossec/bin/wazuh-modulesd
           ├─ 1824 rpm -qa xorg-x11-server*
           └─ 1824 rpm -qa xorg-x11-server*

Sep 12 23:10:53 wazuh-server env[752]: Started wazuh-syscheckd...
Sep 12 23:10:53 wazuh-server env[752]: wazuh-remoted: Process 1534 not used by Wazuh, removing...
Sep 12 23:10:54 wazuh-server env[752]: Started wazuh-remoted...
Sep 12 23:10:54 wazuh-server env[752]: wazuh-logcollector: Process 1604 not used by Wazuh, re...g...
Sep 12 23:10:54 wazuh-server env[752]: Started wazuh-logcollector...
Sep 12 23:10:54 wazuh-server env[752]: wazuh-monitorfd: Process 1648 not used by Wazuh, removing...
Sep 12 23:10:54 wazuh-server env[752]: Started wazuh-monitorfd...
Sep 12 23:10:55 wazuh-server env[752]: wazuh-modulesd: Process 1663 not used by Wazuh, removing...
Sep 12 23:10:56 wazuh-server env[752]: Started wazuh-modulesd...
Sep 12 23:10:58 wazuh-server env[752]: Completed.
Hint: Some lines were ellipsized, use -l to show in full.
[root@wazuh-server ~]#
```

- Check the status of the wazuh-indexer. To check run the command “# systemctl status wazuh-indexer”)

```
[root@wazuh-server ~]# systemctl status wazuh-indexer
● wazuh-indexer.service - Wazuh-indexer
   Loaded: loaded (/usr/lib/systemd/system/wazuh-indexer.service; enabled; vendor preset: disabled)
   Active: failed (Result: timeout) since Mon 2022-09-12 17:41:11 UTC; 1min 33s ago
     Docs: https://documentation.wazuh.com
   Process: 749 ExecStart=/usr/share/wazuh-indexer/bin/systemd-entrypoint -p ${PID_DIR}/wazuh-indexer.pid --quiet (code=exited, status=143)
   Main PID: 749 (code=exited, status=143)
[root@wazuh-server ~]#
```

- If the wazuh-indexer is not running then enable and start the wazuh-indexer service.

Command:

- 1.Systemctl enable wazuh-indexer
- 2.Systemctl start wazuh-indexer
- 3.Systemctl status wazuh-indexer

```
[root@wazuh-server ~]# systemctl start wazuh-indexer
[root@wazuh-server ~]# systemctl status wazuh-indexer
● wazuh-indexer.service - Wazuh-indexer
   Loaded: loaded (/usr/lib/systemd/system/wazuh-indexer.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2022-09-12 17:44:02 UTC; 7s ago
     Docs: https://documentation.wazuh.com
   Main PID: 2675 (java)
      CGroup: /system.slice/wazuh-indexer.service
              └─2675 /usr/share/wazuh-indexer/jdk/bin/java -Xshare:auto -Dopensearch.networkaddress....
```

Sep 12 17:43:54 wazuh-server systemd-entrypoint[2675]: WARNING: An illegal reflective access op...ed  
Sep 12 17:43:54 wazuh-server systemd-entrypoint[2675]: WARNING: Illegal reflective access by io...se  
Sep 12 17:43:54 wazuh-server systemd-entrypoint[2675]: WARNING: Please consider reporting this ...ma  
Sep 12 17:43:54 wazuh-server systemd-entrypoint[2675]: WARNING: Use --illegal-access=warn to en...ns  
Sep 12 17:43:54 wazuh-server systemd-entrypoint[2675]: WARNING: All illegal access operations w...se  
Hint: Some lines were ellipsized, use -l to show in full.
[root@wazuh-server ~]#

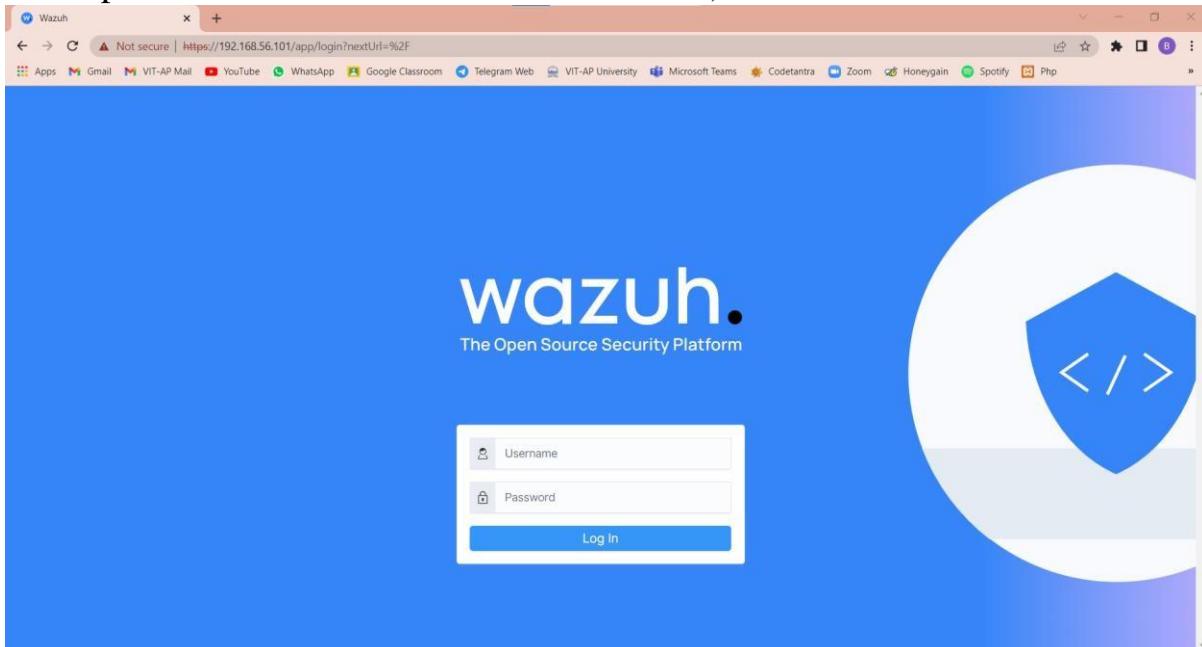
#### ❖ Some useful commands for wazuh server:

<b>start</b>	Start the Wazuh processes.	
<b>stop</b>	Stop the Wazuh processes.	
<b>restart</b>	Restart the Wazuh processes.	
<b>reload</b>	Restart all Wazuh processes except wazuh-execd. This allows an agent to reload without losing active response status. This option is not available on a local Wazuh installation.	
<b>status</b>	Determine which Wazuh processes are running.	
<b>info</b>	Prints the Wazuh installation type, version, and revision in environment variables format.	
<b>info</b>	<b>[-v -r -t]</b>	Only one option at the time, prints the value of: version, revision or type.
<b>enable</b>	<b>debug</b>	Run all Wazuh daemons in debug mode.
<b>disable</b>	<b>debug</b>	Turn off debug mode.

➤ Return back to the step 7. Now again processed to the wauh-dashboard in then chrome.

A dashboard will be opened enter the credentials

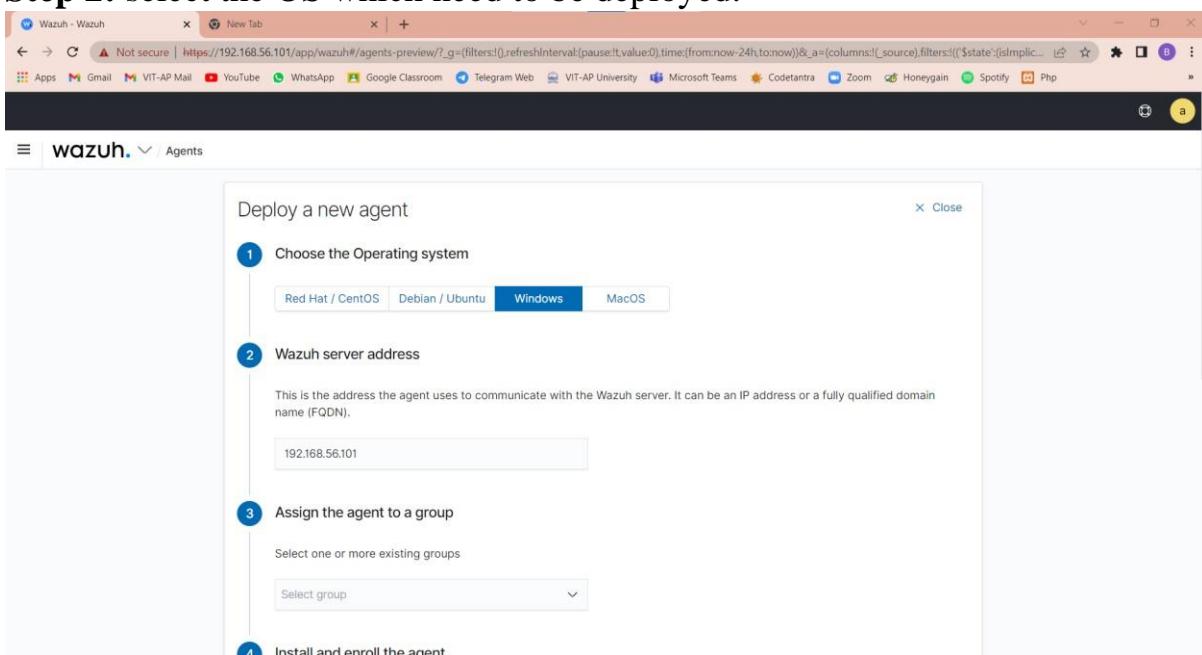
- Username: admin
- Password: admin (By default the wauh-server will have the password and the username as the admin)



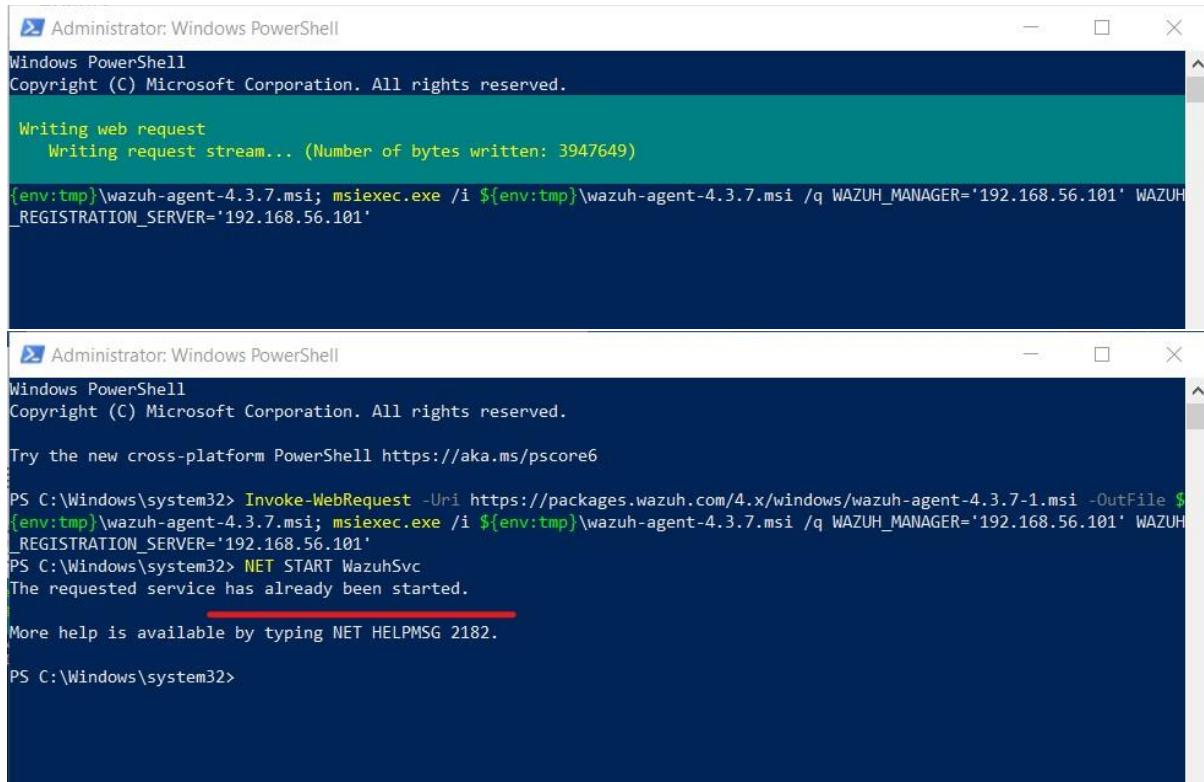
## Deploying the agents to monitor the security activity.

**Step 1:** Select add agents

**Step 2:** select the OS which need to be deployed.



## Step 2: Open the Windows Power Shell with administrator and the link that are in step 4 and 5.



The image contains two side-by-side screenshots of Windows PowerShell windows. Both windows have the title bar "Administrator: Windows PowerShell".

The top window shows command-line output:

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

Writing web request
Writing request stream... (Number of bytes written: 3947649)

{env:tmp}\wazuh-agent-4.3.7.msi; msieexec.exe /i ${env:tmp}\wazuh-agent-4.3.7.msi /q WAZUH_MANAGER='192.168.56.101' WAZUH_REGISTRATION_SERVER='192.168.56.101'
```

The bottom window shows command-line output:

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

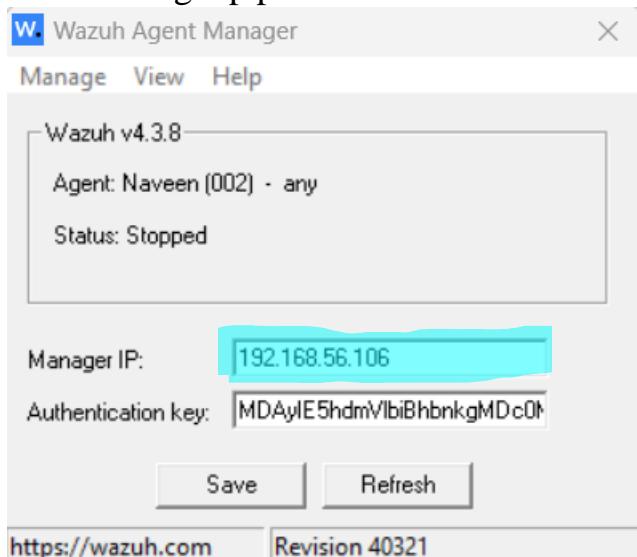
PS C:\Windows\system32> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.3.7-1.msi -OutFile ${env:tmp}\wazuh-agent-4.3.7.msi; msieexec.exe /i ${env:tmp}\wazuh-agent-4.3.7.msi /q WAZUH_MANAGER='192.168.56.101' WAZUH_REGISTRATION_SERVER='192.168.56.101'
PS C:\Windows\system32> NET START WazuhSvc
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.

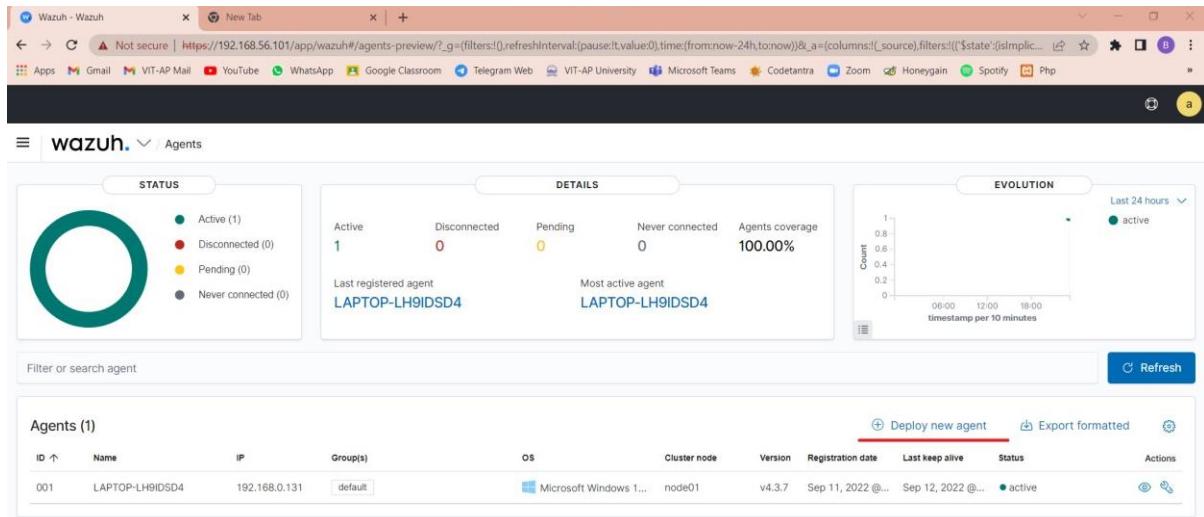
PS C:\Windows\system32>
```

Alternative way to add the agents.

- Install the wazuh agent in the agents system and run the ip of the wazuh-server ,which automatically deploy the agents in the wazuh-server.
- Under mange ip provide the wazuh-server ip.



➤ Sample Dashboard of the Wazuh-server.



- In wazuh-server we can monitor alerts, agents, and API traffic. We can also set up custom alert rules and monitor specific log files.
- There are many things that we can monitor in Wazuh server, including:
  - The number of events that have been processed
  - The number of alerts that have been generated
  - The number of agents that are connected
  - The average response time for agents
  - The load on the server

## Suricata Configuration with Ubuntu 22.04

### What is Suricata?

Suricata is a high performance, open-source network analysis and threat detection software used by most private and public organizations, and embedded by major vendors to protect their assets.

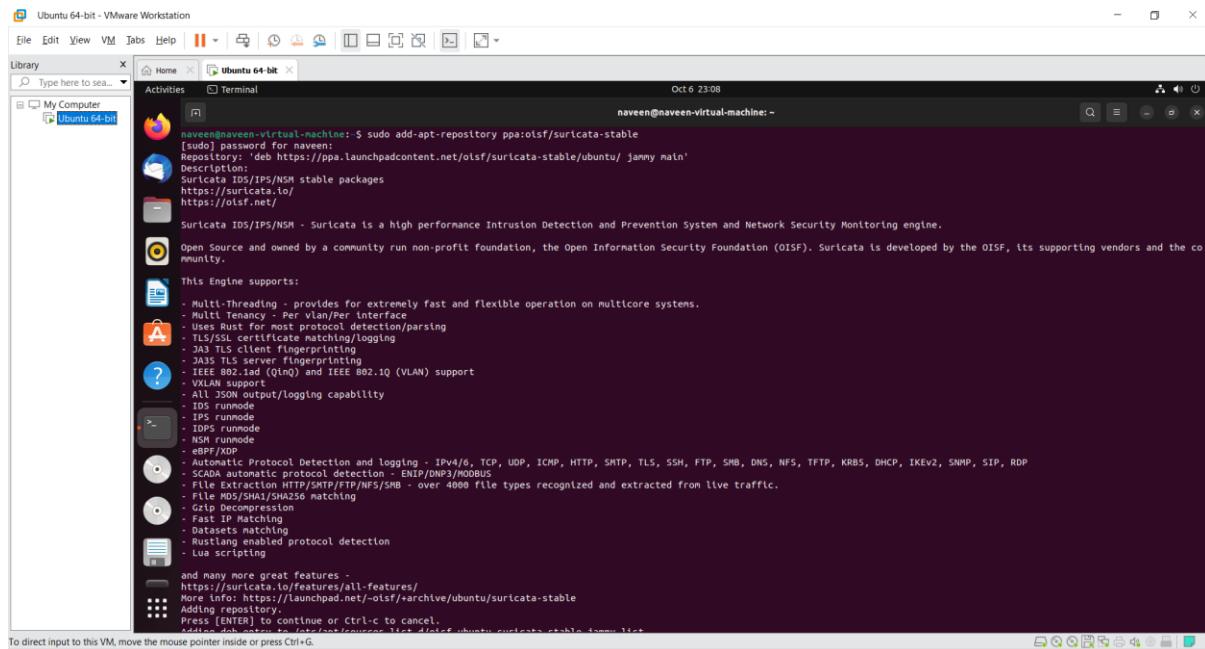
### Use of Suricata?

intrusion detection system (IDS) and an intrusion prevention system (IPS)

Installation:

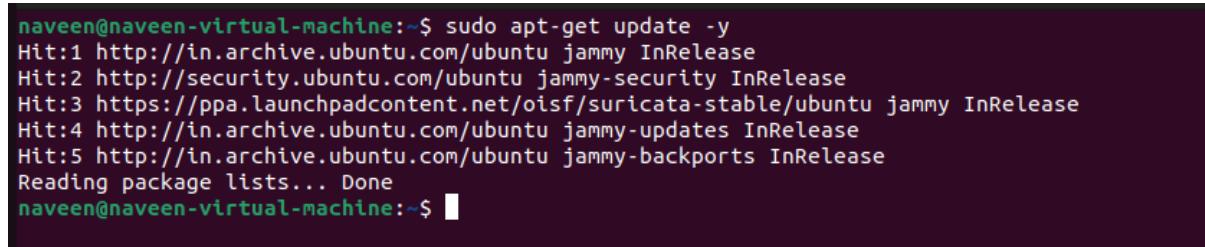
**Step 1:** Open the ubuntu terminal

**Step 2:** Add the Suricata repository by using the following command “**sudo add-apt-repository ppa:oisf/suricata-stable**”



```
naveen@naveen-virtual-machine:~$ sudo add-apt-repository ppa:oisf/suricata-stable
[sudo] password for naveen:
Repository: 'deb https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu/ jammy main'
Description:
Suricata IDS/IPS/NSM stable packages
https://suricata.io/
https://oisf.net/
Suricata IDS/IPS/NSM - Suricata is a high performance Intrusion Detection and Prevention System and Network Security Monitoring engine.
Open Source and owned by a community run non-profit foundation, the Open Information Security Foundation (OISF). Suricata is developed by the OISF, its supporting vendors and the community.
This Engine supports:
- Multi-Threading - provides for extremely fast and flexible operation on multicore systems.
- Multi Tenancy - Per vlan/Per Interface
- Uses Rust for most protocol detection/parsing
- TLS/SSL certificate matching/logging
- JA3 TLS client fingerprinting
- JA3S TLS server fingerprinting
- IEEE 802.1Q (VLAN) and IEEE 802.10 (VLAN) support
- VXLAN support
- All JSON output/logging capability
- IDS runmode
- IPS runmode
- DDoS runmode
- NSM runmode
- eBPF/XDP
- Automatic Protocol Detection and logging - IPv4/6, TCP, UDP, ICMP, HTTP, SMTP, TLS, SSH, SMB, DNS, NFS, TFTP, KRBS, DHCP, IKEv2, SNMP, SIP, RDP
- SCADA automatic protocol detection - ENIP/DNP3/MODBUS
- 2.5Gbps+吞吐量 (TCP/UDP/IPv4/IPv6/SMB)
- File MD5/SHA1/SHA256 matching
- Gzip Decompression
- Fast IP Matching
- Datasets matching
- Rustlang enabled protocol detection
- Lua scripting
and many more great features -
https://suricata.io/features/all-features/
https://suricata.io/launchpad.net-oisf/+archive/ubuntu/suricata-stable
Adding repository...
Press [ENTER] to continue or Ctrl-c to cancel.
Addition done successfully.
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

**Step 3:** Update the terminal by running the command “ **sudo apt-get update -y** ”



```
naveen@naveen-virtual-machine:~$ sudo apt-get update -y
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu jammy InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
naveen@naveen-virtual-machine:~$
```

#### **Step 4:** Install the suricata by using the following command “`sudo apt-get install suricata`”

```
naveen@naveen-virtual-machine:~$ sudo apt-get install suricata
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaaacs0 libaoam3 libass9 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas3
libbluray2 libbsz2b0 libchromaprint1 libcodecs2-1.0 libdavids libflite1 libgme0 libgsml1 libgststreamer-plugins-bad1.0-0 libgdgm12 libl1v-0 libmfx1 libmysofa1 libnorm1
libopenmp0 libppm-5.3-0 libpostproc55 librabbitmq4 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libssord-0-0 librato1.4-gnutls libssh-gcrypt-4 libswresample3
libwscales libvdread0 libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers
mesa-vdpau-drivers pocketsphinx-en-us systemd-hwe-hwdb va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libevent-core-2.1-7 libevent-pthreads-2.1-7 libhiredis0.14 libhttp2 libhyperscan5 libluajit-5.1-2 libluajit-5.1-common liblzma-dev libnet1 libnetfilter-queue1
Suggested packages:
liblzma-doc
The following NEW packages will be installed:
libevent-core-2.1-7 libevent-pthreads-2.1-7 libhiredis0.14 libhttp2 libhyperscan5 libluajit-5.1-2 libluajit-5.1-common liblzma-dev libnet1 libnetfilter-queue1 suricata
0 upgraded, 11 newly installed, 0 to remove and 150 not upgraded.
Need to get 5,412 kB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

#### **Step 5:** After installing the Suricata successfully .Let's start the suricata service by running the following commands.

-“`sudo systemctl enable suricate.service` ”

```
naveen@naveen-virtual-machine:~$ sudo systemctl enable suricata.service
suricata.service is not a native service, redirecting to systemd-sysv-install.
```

-“`sudo systemctl start suricata.service` ”

-“`sudo systemctl status suricata.service` ”

```
naveen@naveen-virtual-machine:~$ sudo systemctl status suricata.service
● suricata.service - LSB: Next Generation IDS/IPS
  Loaded: loaded (/etc/init.d/suricata; generated)
  Active: active (exited) since Thu 2022-10-06 23:13:08 IST; 1min 55s ago
    Docs: man:systemd-sysv-generator(8)
   CPU: 159ms

Oct 06 23:13:08 naveen-virtual-machine systemd[1]: Starting LSB: Next Generation IDS/IPS...
Oct 06 23:13:08 naveen-virtual-machine suricata[4346]: Starting suricata in IDS (af-packet) mode... done.
Oct 06 23:13:08 naveen-virtual-machine systemd[1]: Started LSB: Next Generation IDS/IPS.
naveen@naveen-virtual-machine:~$
```

#### **Step 6:** At the time of the modification or the configuration Suricata service should be stopped to avoid misconfiguration. To stop the service run the following command”`sudo systemctl stop suricata.service` ”

```
naveen@naveen-virtual-machine:~$ sudo systemctl stop suricata
naveen@naveen-virtual-machine:~$ sudo systemctl status suricata.service
× suricata.service - LSB: Next Generation IDS/IPS
  Loaded: loaded (/etc/init.d/suricata; generated)
  Active: failed (Result: exit-code) since Thu 2022-10-06 23:18:30 IST; 3s ago
    Docs: man:systemd-sysv-generator(8)
   Process: 4830 ExecStop=/etc/init.d/suricata stop (code=exited, status=1/FAILURE)
   CPU: 21ms

Oct 06 23:13:08 naveen-virtual-machine systemd[1]: Starting LSB: Next Generation IDS/IPS...
Oct 06 23:13:08 naveen-virtual-machine suricata[4346]: Starting suricata in IDS (af-packet) mode... done.
Oct 06 23:13:08 naveen-virtual-machine systemd[1]: Started LSB: Next Generation IDS/IPS.
Oct 06 23:18:30 naveen-virtual-machine systemd[1]: Stopping LSB: Next Generation IDS/IPS...
Oct 06 23:18:30 naveen-virtual-machine suricata[4830]: Stopping suricata: /etc/init.d/suricata: 119: kill: No such process
Oct 06 23:18:30 naveen-virtual-machine systemd[1]: suricata.service: Control process exited, code=exited, status=1/FAILURE
Oct 06 23:18:30 naveen-virtual-machine systemd[1]: suricata.service: Failed with result 'exit-code'.
Oct 06 23:18:30 naveen-virtual-machine systemd[1]: Stopped LSB: Next Generation IDS/IPS.
naveen@naveen-virtual-machine:~$
```

- The files and configuration file be stored in suricata “ ls -al /etc/suricata/ ”

```
naveen@naveen-virtual-machine:~$ ls -al /etc/suricata/
total 108
drwxr-xr-x  3 root root  4096 Oct  6 23:13 .
drwxr-xr-x 131 root root 12288 Oct  6 23:13 ..
-rw-r--r--  1 root root  3327 Sep 27 23:28 classification.config
-rw-r--r--  1 root root 1375 Sep 27 23:28 reference.config
drwxr-xr-x  2 root root  4096 Oct  6 23:13 rules
-rw-r--r--  1 root root 74802 Sep 28 16:58 suricata.yaml
-rw-r--r--  1 root root 1644 Sep 27 23:28 threshold.config
naveen@naveen-virtual-machine:~$
```

- The default rules of the suricata will be stored in the path “ ls -al /etc/suricata/rules ”

```
naveen@naveen-virtual-machine:~$ ls -al /etc/suricata/rules
total 140
drwxr-xr-x 2 root root  4096 Oct  6 23:13 .
drwxr-xr-x 3 root root  4096 Oct  6 23:13 ..
-rw-r--r-- 1 root root 1858 Sep 27 23:28 app-layer-events.rules
-rw-r--r-- 1 root root 20821 Sep 27 23:31 decoder-events.rules
-rw-r--r-- 1 root root  468 Sep 27 23:28 dhcp-events.rules
-rw-r--r-- 1 root root 1221 Sep 27 23:28 dnp3-events.rules
-rw-r--r-- 1 root root 1041 Sep 27 23:28 dns-events.rules
-rw-r--r-- 1 root root 4003 Sep 27 23:28 files.rules
-rw-r--r-- 1 root root 2128 Sep 27 23:31 http2-events.rules
-rw-r--r-- 1 root root 13390 Sep 27 23:31 http-events.rules
-rw-r--r-- 1 root root 2717 Sep 27 23:31 ipsec-events.rules
-rw-r--r-- 1 root root  585 Sep 27 23:28 kerberos-events.rules
-rw-r--r-- 1 root root 2078 Sep 27 23:28 modbus-events.rules
-rw-r--r-- 1 root root 2187 Sep 27 23:28 mqtt-events.rules
-rw-r--r-- 1 root root  558 Sep 27 23:31 nfs-events.rules
-rw-r--r-- 1 root root  558 Sep 27 23:28 ntp-events.rules
-rw-r--r-- 1 root root 4346 Sep 27 23:28 smb-events.rules
-rw-r--r-- 1 root root 5167 Sep 27 23:31 smtp-events.rules
-rw-r--r-- 1 root root  719 Sep 27 23:28 ssh-events.rules
-rw-r--r-- 1 root root 12992 Sep 27 23:31 stream-events.rules
-rw-r--r-- 1 root root 6861 Sep 27 23:28 tls-events.rules
naveen@naveen-virtual-machine:~$
```

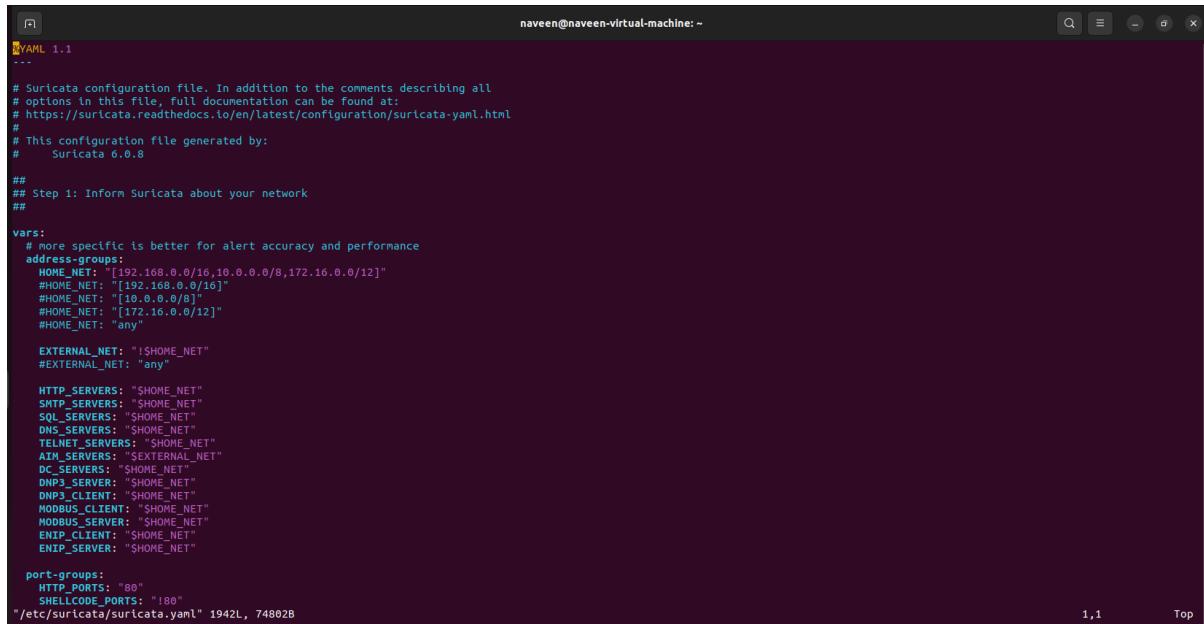
## Step 7: Install the vim in the Ubuntu by running the following command

“ sudo apt-get install vim -y ”

```
naveen@naveen-virtual-machine:~$ sudo apt-get install vim -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgmp3c2 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas3
libbluray2 libbs2b0 libchromaprint1 libcodecs2-1.0 libdavids libflite1 libgnome libgstreamer-plugins-bad1.0-0 liblqdmm12 liblly1 libmysofa1 libnorm1
libopennpt0 libppm-5.3-0 libpostproc5 librabbitmq4 librubberband2 libserd-0.0 libshine3 libsnappy1v5 libzord-0.0 libsrat0-0.0 libst1.4-gnutls libssh-gcrypt-4 libswresample3
libwscales libxfreadd libdrm2 libva-wayland2 libva-x11-2 libva2 libvdpaui libvidstab1.i libx265-199 libxvidcore4 libzimg2 libzvbi-common libzvbi0 mesa-va-drivers
mesa-vdpau-drivers pocketsphinx-en-us systemd-hwdb va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  vim-common vim-runtime vim-tiny
Suggested packages:
  ctags vim-doc vim-scripts indent
The following NEW packages will be installed:
  vim vim-runtime
The following packages will be upgraded:
  vim-common vim-tiny
```

**Step 8:** Run the following command after installing the vim package in the ubuntu to make the changes in configuration.

“ sudo vim /etc/suricata/suricata.yaml ”



```
YAML 1.1
...
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://suricata.readthedocs.io/en/latest/configuration/suricata-yaml.html
#
# This configuration file generated by:
#   Suricata 6.0.8

## Step 1: Inform Suricata about your network
##


vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: ["192.168.0.0/16,10.0.0.0/8,172.16.0.0/12"]
    #HOME_NET: ["192.168.0.0/16"]
    #HOME_NET: ["10.0.0.0/8"]
    #HOME_NET: ["172.16.0.0/12"]
    #HOME_NET: "any"

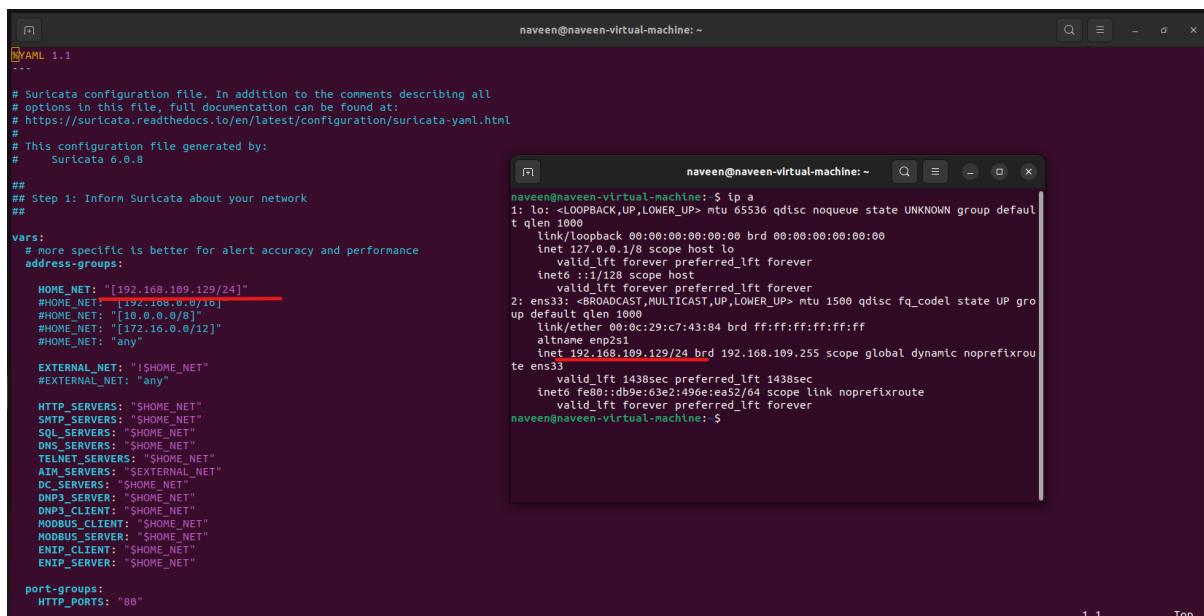
    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "SHOME_NET"
    SMTP_SERVERS: "SHOME_NET"
    SQL_SERVERS: "SHOME_NET"
    DNS_SERVERS: "SHOME_NET"
    TELNET_SERVERS: "SHOME_NET"
    AIM_SERVERS: "SEXTERNAL_NET"
    DC_SERVERS: "SHOME_NET"
    DNP3_SERVER: "SHOME_NET"
    DNP3_CLIENT: "SHOME_NET"
    MODBUS_CLIENT: "SHOME_NET"
    MODBUS_SERVER: "SHOME_NET"
    ENIP_CLIENT: "SHOME_NET"
    ENIP_SERVER: "SHOME_NET"

port-groups:
  HTTP_PORTS: "80"
  SHELLCODE_PORTS: "180"
"/etc/suricata/suricata.yaml" 1942L, 74802B
```

- ❖ For searching a word in yaml file use “/”
- ❖ For inserting the data in yaml file use “ctrl+o”
- ❖ For saving the data in yaml file use “esc, ctrl+: , wq”
- ❖ For quitting from a yaml file use “:q”

**Step 9:** Change the HOME\_NET details to you laptop's ip with subnet range  
(you can get the HOME\_NET ip by using the above command “ip a”)



```
YAML 1.1
...
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://suricata.readthedocs.io/en/latest/configuration/suricata-yaml.html
#
# This configuration file generated by:
#   Suricata 6.0.8

## Step 1: Inform Suricata about your network
##


vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: ["192.168.109.129/24"] -----
    #HOME_NET: ["192.168.0.0/16"]
    #HOME_NET: ["10.0.0.0/8"]
    #HOME_NET: ["172.16.0.0/12"]
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "SHOME_NET"
    SMTP_SERVERS: "SHOME_NET"
    SQL_SERVERS: "SHOME_NET"
    DNS_SERVERS: "SHOME_NET"
    TELNET_SERVERS: "SHOME_NET"
    AIM_SERVERS: "SEXTERNAL_NET"
    DC_SERVERS: "SHOME_NET"
    DNP3_SERVER: "SHOME_NET"
    DNP3_CLIENT: "SHOME_NET"
    MODBUS_CLIENT: "SHOME_NET"
    MODBUS_SERVER: "SHOME_NET"
    ENIP_CLIENT: "SHOME_NET"
    ENIP_SERVER: "SHOME_NET"

port-groups:
  HTTP_PORTS: "80"
  SHELLCODE_PORTS: "180"
```

```
naveen@naveen-virtual-machine: ~ $ 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 brd 127.255.255.255 scope host lo
      valid_lft forever preferred_lft forever
      inet6 ::1/128 brd :: scope host lo
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group 0
  qlen 1000
    link/ether 00:0c:29:c7:43:84 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.109.129/24 brd 192.168.109.255 scope global dynamic noprefixroute
      valid_lft 1438sec preferred_lft 1438sec
      inet6 fe80::db9e:63e2:496e:a52/64 brd fe80::ff:ff:ff:ff:ff:ff scope link noprefixroute
        valid_lft forever preferred_lft forever
naveen@naveen-virtual-machine: ~
```

## Step 10: Search for “af-packet” in yaml and change the interface name according to your internet interface:

The screenshot shows two terminal windows side-by-side. The left terminal window displays the Suricata configuration file (suricata.yaml) with the 'af-packet' section highlighted. The right terminal window shows the output of the 'ifconfig' command, listing the network interfaces. The 'ens3' interface is selected and highlighted in red.

```

naveen@naveen-virtual-machine: ~
# type: json
- file:
  enabled: yes
  level: info
  filename: suricata.log
  # type: json
- syslog:
  enabled: no
  facility: local5
  format: [%{!k} <%d> -- "]
  # type: json

##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# Linux high speed capture support
af-packet:
- interface: ens3
  # Number of receive threads. "auto" uses the number of cores
  #threads: auto
# Default clusterid. AF_PACKET will load balance packets based on flow.
cluster-idx: 99
# This is only supported for Linux kernel > 3.1
# Possible value are:
# * cluster_flow: all packets of a given flow are sent to the same socket
# * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
# * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
# socket. Requires at least Linux 3.14.
# * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
# more info.
# Recommended nodes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
# with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
cluster-type: cluster_flow
# In some fragmentation cases, the hash can not be computed. If "defrag" is set
# to yes, the kernel will do the needed defragmentation before sending the packets.
defrag: yes
# To use the ring feature of AF_PACKET, set 'use-mmap' to yes
-- INSERT --

```

```

naveen@naveen-virtual-machine: ~
Unpacking net-tools (1.60-git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60-git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
naveen@naveen-virtual-machine: ~$ ifconfig
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.109.129 netmask 255.255.255.0 broadcast 192.168.109.255
inet6 fe80::db9e:63e2:496e:ea52 prefixlen 64 scoprid 0x20<link>
ether 00:0c:29:c7:43:81 txqueuelen 1000 (Ethernet)
RX packets 944 bytes 308446 (308.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 551 bytes 56592 (56.5 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scoprid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 300 bytes 27408 (27.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 308 bytes 27408 (27.4 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
naveen@naveen-virtual-machine: ~

```

## Step 11: Search for “pcap” in yaml file and make a change in interface

The screenshot shows a single terminal window displaying the Suricata configuration file (suricata.yaml). The 'pcap' section is highlighted. The configuration includes details about buffer size, checksum validation, and interface selection.

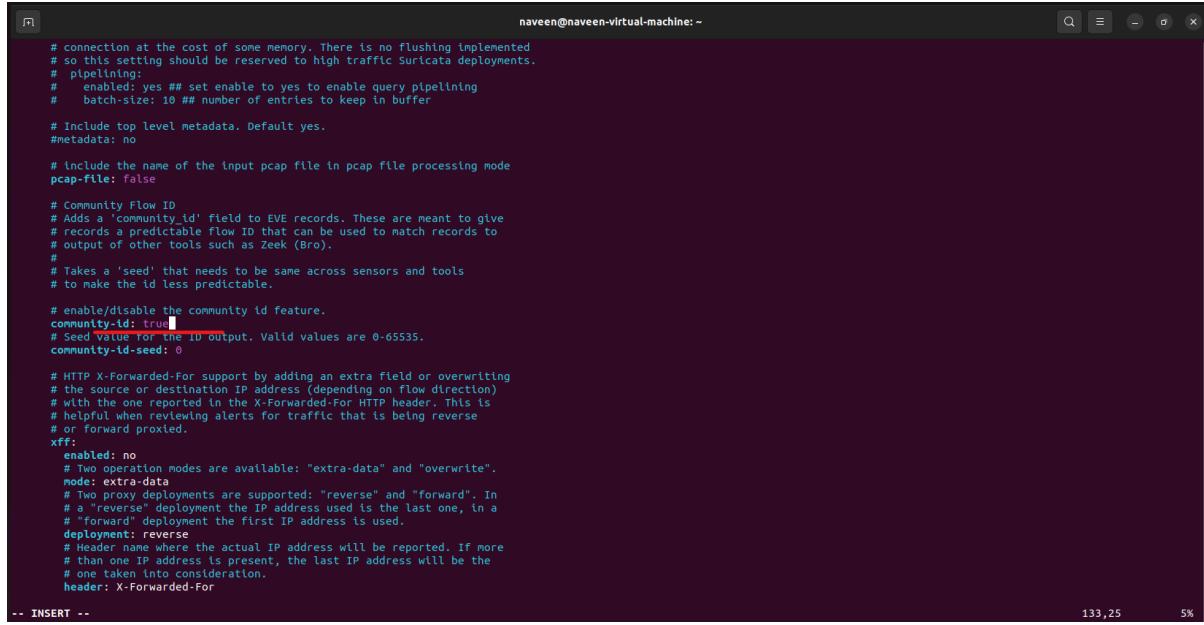
```

naveen@naveen-virtual-machine: ~
# recv buffer size, increased value could improve performance
# buffer-size: 32768
# Set to yes to disable promiscuous mode
# disable-promisc: no
# Choose checksum verification mode for the interface. At the moment
# of the capture, some packets may have an invalid checksum due to
# the checksum computation being offloaded to the network card.
# Possible values are:
# - kernel: use indication sent by kernel for each packet (default)
# - yes: checksum validation is forced
# - no: checksum validation is disabled
# - auto: Suricata uses a statistical approach to detect when
# checksum off-loading is used.
# Warning: 'capture.checksum-validation' must be set to yes to have any validation
#checksum-checks: kernel
# BPF filter to apply to this interface. The pcap filter syntax applies here.
#bpf-filters: port 80 or udp
# You can use the following variables to activate AF_PACKET tap or IPS mode.
# If copy-mode is set to lps or tap, the traffic coming to the current
# interface will be copied to the copy-iface interface. If 'tap' is set, the
# copy is complete. If 'lps' is set, the packet matching the 'drop' action
# will not be copied.
#copy-mode: tap
#copy-iface: eth1
# For eBPF and XDP setup including bypass, filter and load balancing, please
# see doc/userguide/capture-hardware/ebpf-xdp.rst for more info.
# Put default values here. These will be used for an interface that is not
# in the list above.
- interface: default
  #threads: auto
  #use-mmap: no
  #packet-v3: yes

# Cross platform libpcap capture support
pcap:
- interface: ns33
  # By default, we will try to use mmap capture and will use "buffer-size"
  # as total memory used by the ring. So set this to something bigger
  # than 1% of your bandwidth.
  #buffer-size: 16777216
  #bpf-filters: "tcp and port 25"
  # Choose checksum verification mode for the interface. At the moment
-- INSERT --

```

## Step 12: Under pcap-file set the community-id as true:



```
naveen@naveen-virtual-machine: ~
# connection at the cost of some memory. There is no flushing implemented
# so this setting should be reserved to high traffic Suricata deployments.
# pipelining:
#   enabled: yes ## set enable to yes to enable query pipelining
#   batch-size: 10 ## number of entries to keep in buffer

# Include top level metadata. Default yes.
#metadata: no

# Include the name of the input pcap file in pcap file processing mode
pcap-file: false

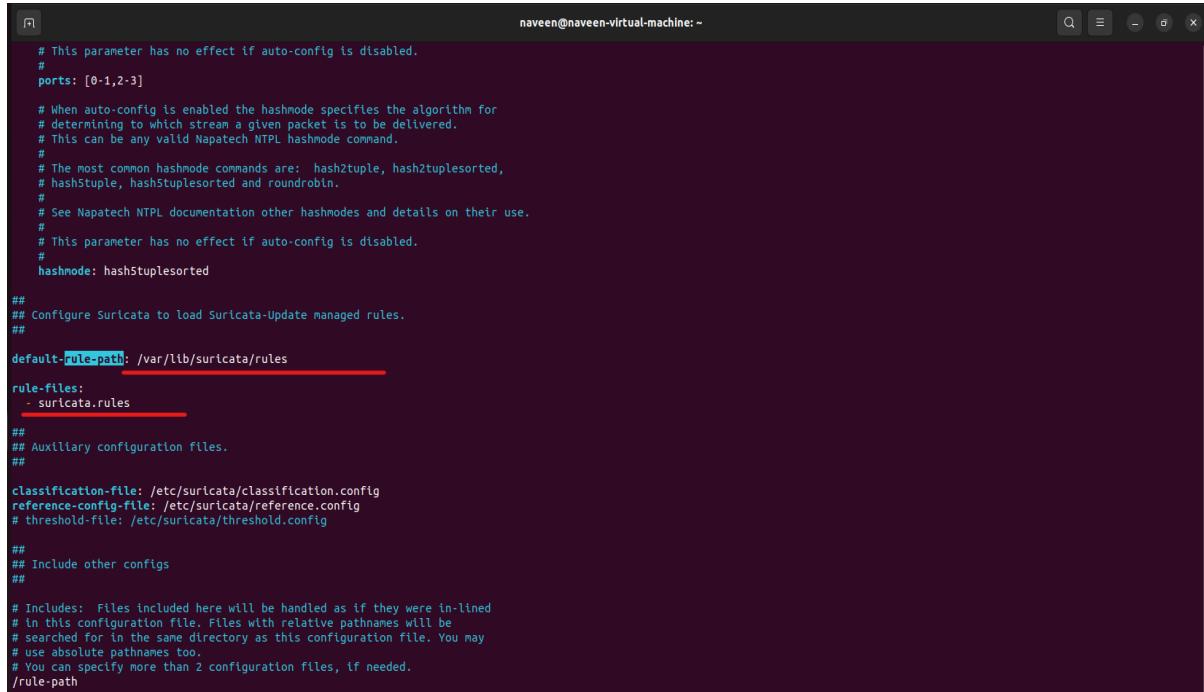
# Community Flow ID
# Adds a 'community_id' field to EVE records. These are meant to give
# records a predictable flow ID that can be used to match records to
# output of other tools such as Zeek (Bro).
#
# Takes a 'seed' that needs to be same across sensors and tools
# to make the id less predictable.

# enable/disable the community id feature.
community-id: true
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 0

# HTTP X-Forwarded-For support by adding an extra field or overwriting
# the source or destination IP address (depending on flow direction)
# with the one reported in the X-Forwarded-For HTTP header. This is
# helpful when reviewing alerts for traffic that is being reverse
# or forward proxied.
xff:
  enabled: no
  # Two operation modes are available: "extra-data" and "overwrite".
  mode: extra-data
  # Two proxy deployments are supported: "reverse" and "forward". In
  # a "reverse" deployment the IP address used is the last one, in a
  # "forward" deployment the first IP address is used.
  deployment: reverse
  # Header name where the actual IP address will be reported. If more
  # than one IP address is present, the last IP address will be the
  # one taken into consideration.
  header: X-Forwarded-For

-- INSERT --
133,25 5%
```

## Step 13: Search for the rule-path and make sure that above details are present:



```
naveen@naveen-virtual-machine: ~
# This parameter has no effect if auto-config is disabled.
#
ports: [0-1,2-3]

# When auto-config is enabled the hashmode specifies the algorithm for
# determining to which stream a given packet is to be delivered.
# This can be any valid Napatech NTPL hashmode command.
#
# The most common hashmode commands are: hash2tuple, hash2tuplesorted,
# hashStuple, hashStuplesorted and roundrobin.
#
# See Napatech NTPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hashStuplesorted

## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules
rule-files:
  - suricata.rules
##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config

##
## Include other configs
##

# Includes: Files included here will be handled as if they were in-lined
# in this configuration file. Files with relative pathnames will be
# searched for in the same directory as this configuration file. You may
# use absolute pathnames too.
# You can specify more than 2 configuration files, if needed.
/rule-path
```

## Step 14: Update the suricata file for seeing the by default rule files: (skip the warning if you got it)

```

naveen@naveen-virtual-machine:~$ sudo suricata-update
7/10/2022 -- 06:51:32 - <Info> -- Using data-directory /var/lib/suricata.
7/10/2022 -- 06:51:32 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
7/10/2022 -- 06:51:32 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
7/10/2022 -- 06:51:32 - <Info> -- Found Suricata version 6.0.8 at /usr/bin/suricata.
7/10/2022 -- 06:51:32 - <Info> -- Loading /etc/suricata/suricata.yaml
7/10/2022 -- 06:51:32 - <Info> -- Disabling rules for protocol http2
7/10/2022 -- 06:51:32 - <Info> -- Disabling rules for protocol modbus
7/10/2022 -- 06:51:32 - <Info> -- Disabling rules for protocol dnp3
7/10/2022 -- 06:51:32 - <Info> -- Disabling rules for protocol entp
7/10/2022 -- 06:51:32 - <Info> -- No sources configured, will use Emerging Threats Open
7/10/2022 -- 06:51:32 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-6.0.8/emerging.rules.tar.gz.
100% - 3486076/3486076
7/10/2022 -- 06:51:35 - <Info> -- Done.
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/app-layer-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/decoder-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/dns-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/dns-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/ffiles.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/http-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/https-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/kerberos-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/modbus-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/nfs-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/ntp-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/smb-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/smtp-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/stream-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Loading distribution rule file /etc/suricata/rules/tls-events.rules
7/10/2022 -- 06:51:35 - <Info> -- Ignoring file rules/emerging-deleted.rules
7/10/2022 -- 06:51:38 - <Info> -- Ignoring file rules/suricata.rules.
7/10/2022 -- 06:51:38 - <Info> -- Disabled 4 rules.
7/10/2022 -- 06:51:38 - <Info> -- Enabled 0 rules.
7/10/2022 -- 06:51:38 - <Info> -- Modified 0 rules.
7/10/2022 -- 06:51:38 - <Info> -- Dropped 0 rules.
7/10/2022 -- 06:51:38 - <Info> -- Enabled 131 rules for flowbit dependencies.
7/10/2022 -- 06:51:38 - <Info> -- Creating directory /var/lib/suricata/rules.
7/10/2022 -- 06:51:38 - <Info> -- Backing up current rules.
7/10/2022 -- 06:51:38 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 35896; enabled: 28332; added: 35896; removed 0; modified: 0
7/10/2022 -- 06:51:39 - <Info> -- Testing with suricata -T.

```

➤ Here is the suricata by default rule files “ sudo ls -al /var/lib/suricata/rules ”

```

naveen@naveen-virtual-machine:~$ sudo ls -al /var/lib/suricata/rules
total 20524
drwxr-x--- 2 root root 4096 Oct  7 06:51 .
drwxr-xr-x  4 root root 4096 Oct  7 06:51 ..
-rw-r--r--  1 root root 3228 Oct  7 06:51 classification.config
-rw-r--r--  1 root root 21001196 Oct  7 06:51 suricata.rules
naveen@naveen-virtual-machine:~$ 

```

➤ If you want to update the suricata rules with other rules that are present in open source use the above command: “sudo suricata-update list-sources”

```

naveen@naveen-virtual-machine:~$ sudo suricata-update list-sources
7/10/2022 -- 06:57:05 - <Info> -- Using data-directory /var/lib/suricata.
7/10/2022 -- 06:57:05 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
7/10/2022 -- 06:57:05 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
7/10/2022 -- 06:57:05 - <Info> -- Found Suricata version 6.0.8 at /usr/bin/suricata.
7/10/2022 -- 06:57:05 - <Info> -- No source index found, running update-sources
7/10/2022 -- 06:57:05 - <Info> -- Downloading https://www.openinfosecfoundation.org/rules/index.yaml
7/10/2022 -- 06:57:07 - <Info> -- Adding all sources
7/10/2022 -- 06:57:07 - <Info> -- Saved /var/lib/suricata/update/cache/index.yaml
Name: et/open
  Vendor: Proofpoint
    Summary: Emerging Threats Open Ruleset
    License: MIT
Name: et/pro
  Vendor: Proofpoint
    Summary: Emerging Threats Pro Ruleset
    License: Commercial
    Replaces: et/open
    Parameters: secret-code
    Subscription: https://www.proofpoint.com/us/threat-insight/et-pro-ruleset
Name: oisf/trafficid
  Vendor: OISF
    Summary: Suricata Traffic ID ruleset
    License: MIT
Name: scwx/enhanced
  Vendor: Secureworks
    Summary: Secureworks suricata-enhanced ruleset
    License: Commercial
    Parameters: secret-code
    Subscription: https://www.secureworks.com/contact/ (Please reference CTU Countermeasures)
Name: scwx/malware
  Vendor: Secureworks
    Summary: Secureworks suricata-malware ruleset
    License: Commercial
    Parameters: secret-code
    Subscription: https://www.secureworks.com/contact/ (Please reference CTU Countermeasures)
Name: scwx/security
  Vendor: Secureworks
    Summary: Secureworks suricata-security ruleset
    License: Commercial
    Parameters: secret-code
    Subscription: https://www.secureworks.com/contact/ (Please reference CTU Countermeasures)

```

- You can use any of the file that was present in list-sources each sources gives different rules (In my case I was adding the malsilo/win-malware)
- For adding it using the above command: “**sudo suricata-update enable-source malsilo/win-malware**”

```
naveen@naveen-virtual-machine:~$ sudo suricata-update enable-source malsilo/win-malware
7/10/2022 -- 07:00:36 - <Info> -- Using data-directory /var/lib/suricata.
7/10/2022 -- 07:00:36 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
7/10/2022 -- 07:00:36 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
7/10/2022 -- 07:00:36 - <Info> -- Found Suricata version 6.0.8 at /usr/bin/suricata.
7/10/2022 -- 07:00:36 - <Info> -- Creating directory /var/lib/suricata/update/sources
7/10/2022 -- 07:00:36 - <Info> -- Enabling default source et/open
7/10/2022 -- 07:00:36 - <Info> -- Source malsilo/win-malware enabled
naveen@naveen-virtual-machine:~$
```

- Try to update the suricata (sudo suricata-update) and make sure that there is no error.

Test the suricata file by using the above command: “**sudo suricata -T -c /etc/suricata/suricata.yaml -v**” and make sure that there is no error.

```
naveen@naveen-virtual-machine:~$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
7/10/2022 -- 07:03:52 - <Info> - Running suricata under test mode
7/10/2022 -- 07:03:52 - <Notice> - This is Suricata version 6.0.8 RELEASE running in SYSTEM mode
7/10/2022 -- 07:03:52 - <Info> - CPU(s)/cores online: 4
7/10/2022 -- 07:03:52 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol stp enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
7/10/2022 -- 07:03:52 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol mgmt enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
7/10/2022 -- 07:03:52 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol rdp enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
7/10/2022 -- 07:03:53 - <Info> - fast output device (regular) initialized: fast.log
7/10/2022 -- 07:03:53 - <Info> - eve-log output device (regular) initialized: eve.json
7/10/2022 -- 07:03:53 - <Info> - stats output device (regular) initialized: stats.log
7/10/2022 -- 07:04:14 - <Info> - 1 rule files processed. 28768 rules successfully loaded, 0 rules failed
7/10/2022 -- 07:04:14 - <Info> - Threshold config parsed: 0 rule(s) found
7/10/2022 -- 07:04:14 - <Info> - 28771 signatures processed. 1232 are IP-only rules, 5156 are inspecting packet payload, 22156 inspect application layer, 108 are decoder event only
7/10/2022 -- 07:04:25 - <Notice> - Configuration provided has been successfully loaded. Exiting.
7/10/2022 -- 07:04:25 - <Info> - Cleaning up signature grouping structure... complete
naveen@naveen-virtual-machine:~$
```

Step 15: Start the suricata and check the status:

```
naveen@naveen-virtual-machine:~$ sudo systemctl start suricata.service
naveen@naveen-virtual-machine:~$ sudo systemctl status suricata.service
● suricata.service - LSB: Next Generation IDS/IPS
  Loaded: loaded (/etc/init.d/suricata; generated)
  Active: active (exited) since Fri 2022-10-07 06:14:50 IST; 52min ago
    Docs: man:systemd-sysv-generator(8)
  Process: 972 ExecStart=/etc/init.d/suricata start (code=exited, status=0/SUCCESS)
    CPU: 863ms

Oct 07 06:14:50 naveen-virtual-machine systemd[1]: Starting LSB: Next Generation IDS/IPS...
Oct 07 06:14:50 naveen-virtual-machine suricata[972]: Starting suricata in IDS (af-packet) mode... done.
Oct 07 06:14:50 naveen-virtual-machine systemd[1]: Started LSB: Next Generation IDS/IPS.
naveen@naveen-virtual-machine:~$
```

- The log files where the data gets stored “**ls -al /var/log/suricata**”

```
naveen@naveen-virtual-machine:~$ ls -al /var/log/suricata
total 68
drwxr-xr-x  5 root root   4096 Oct  6 23:13 .
drwxrwxr-x 14 root syslog  4096 Oct  7 06:14 ..
drwxr-xr-x  2 root root   4096 Sep 28 16:58 certs
drwxr-xr-x  2 root root   4096 Sep 28 16:58 core
-rw-r--r--  1 root root     0 Oct  6 23:13 eve.json
-rw-r--r--  1 root root     0 Oct  6 23:13 fast.log
drwxr-xr-x  2 root root   4096 Sep 28 16:58 files
-rw-r--r--  1 root root     0 Oct  6 23:13 stats.log
-rw-r--r--  1 root root 43980 Oct  7 07:04 suricata.log
-rw-r--r--  1 root root  2435 Oct  7 06:14 suricata-start.log
```

**Step 16:** Install curl in your system: “`sudo apt-get install curl`”

**Step 17:** Run the repository which is used to test the IDS (Intrusion Detection Services) is working on the network.

- “`curl http://testmyids.org/uid/index.html`

```
root@air-virtual-machine:/home/air# curl http://testmyids.org/uid/index.html
uid=0(root) gid=0(root) groups=0(root)
root@air-virtual-machine:/home/air#
```

**Step 18:** To check whether the IDS logs are running in the background

- “`sudo cat /var/log/suricata/fast.log`”

```
root@air-virtual-machine:/home/air# sudo tail -n1 /var/log/suricata.fast.log
tail: cannot open '/var/log/suricata.fast.log' for reading: No such file or directory
root@air-virtual-machine:/home/air# sudo cat /var/log/suricata/fast.log
10/20/2022-15:07:55.783730  [**] [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely
[Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.152.129:53130 -> 185.125.190
10/20/2022-15:10:12.319609  [**] [1:2013028:7] ET POLICY curl User-Agent Outbound [**] [Classifica
[Priority: 2] {TCP} 192.168.152.129:37150 -> 52.85.234.20:80
10/20/2022-15:10:12.320538  [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [C
fic] [Priority: 2] {TCP} 52.85.234.20:80 -> 192.168.152.129:37150
```

**Step 19:** Now create the rules under the direction of the local rules

Save the file and exit.

- The following command is used to record the ping logs from any system on the network.

```
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping"; sid:1; rev:1;)
```

**Step 20:** Adding the new rules into the suricata rules

-currently we are adding the icmp rule

Command: “`sudo vim /etc/suricata/suricata.yaml`”

Add the rule in the rules path

```
default-rule-path: /var/lib/suricata/rules
rule-files:
  - suricata.rules
  - /etc/suricata/rules/local.rules
```

**Step 21:** Run the command to confirm no errors occurred in the rules set “ sudo suricata -T -c /etc/suricata/suricat.yaml -v “

```
root@air-virtual-machine:/home/air# sudo suricata -T -c /etc/suricata/suricata.yaml -v
20/10/2022 -- 15:37:52 - <Info> - Running suricata under test mode
20/10/2022 -- 15:37:52 - <Notice> - This is Suricata version 6.0.8 RELEASE running in SYSTEM mode
20/10/2022 -- 15:37:52 - <Info> - CPUs/cores online: 2
20/10/2022 -- 15:37:52 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol sip enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/10/2022 -- 15:37:52 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol mqtt enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/10/2022 -- 15:37:52 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol rdp enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/10/2022 -- 15:37:52 - <Info> - fast output device (regular) initialized: fast.log
20/10/2022 -- 15:37:52 - <Info> - eve-log output device (regular) initialized: eve.json
20/10/2022 -- 15:37:52 - <Info> - stats output device (regular) initialized: stats.log
20/10/2022 -- 15:37:57 - <Info> - 2 rule files processed. 29120 rules successfully loaded, 0 rules failed
20/10/2022 -- 15:37:57 - <Info> - Threshold config parsed: 0 rule(s) found
20/10/2022 -- 15:37:57 - <Info> - 29123 signatures processed. 1228 are IP-only rules, 5166 are inspecting packet payload, 22502 inspect application layer, 108 are decoder event only
20/10/2022 -- 15:38:06 - <Notice> - Configuration provided was successfully loaded. Exiting.
20/10/2022 -- 15:38:07 - <Info> - cleaning up signature grouping structure... complete
```

**Step 22:** ping the Ubuntu from same interface network to capture the logs of ping

Check the logs whether the ping is working or not

Command: sudo cat /var/log/suricata/fast.log

Here we pinned from a system with the Ip 192.168.152.1 to the ubuntu having the Ip of 192.168.152.255

```
root@air-virtual-machine:/home/air# sudo cat /var/log/suricata/fast.log
10/20/2022-15:07:55.783730 [**] [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management [**]
[Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.152.129:53130 -> 185.125.190.39:80
10/20/2022-15:10:12.319609 [**] [1:2013028:7] ET POLICY curl User-Agent Outbound [**] [Classification: Attempted Information Leak]
[Priority: 2] {TCP} 192.168.152.129:37150 -> 52.85.234.20:80
10/20/2022-15:10:12.320538 [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic]
[Priority: 2] {TCP} 52.85.234.20:80 -> 192.168.152.129:37150
10/20/2022-15:45:39.615244 [**] [1:1:1] ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.152.1:8 -> 192.168.152.129:0
.129:0
10/20/2022-15:45:39.615294 [**] [1:1:1] ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.152.129:0 -> 192.168.152.1:0
```

- From the above figure it's clearly showing that the logs are detecting by the Suricata which sits on the same network and the logs are displayed immediately when someone pinned the agent in the same network.

## Installing the json in Ubuntu 22.04

- Json is used to read the logs in the GUI format which are produced by the suricate in the Ubuntu

**Step 1:** Install the json by running the following command in ubuntu terminal “  
`sudo apt-get install jq`“

```
52.1.0
root@air-virtual-machine:/home/air# sudo apt-get install jq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libjq1 libonig5
The following NEW packages will be installed:
  jq libjq1 libonig5
0 upgraded, 3 newly installed, 0 to remove and 125 not upgraded.
Need to get 357 kB of archives.
After this operation, 1,087 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libonig5 amd64 6.9.7.1-2build1 [172 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libjq1 amd64 1.6-2.1ubuntu3 [133 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 jq amd64 1.6-2.1ubuntu3 [52.5 kB]
Fetched 357 kB in 4s (98.7 kB/s)
```

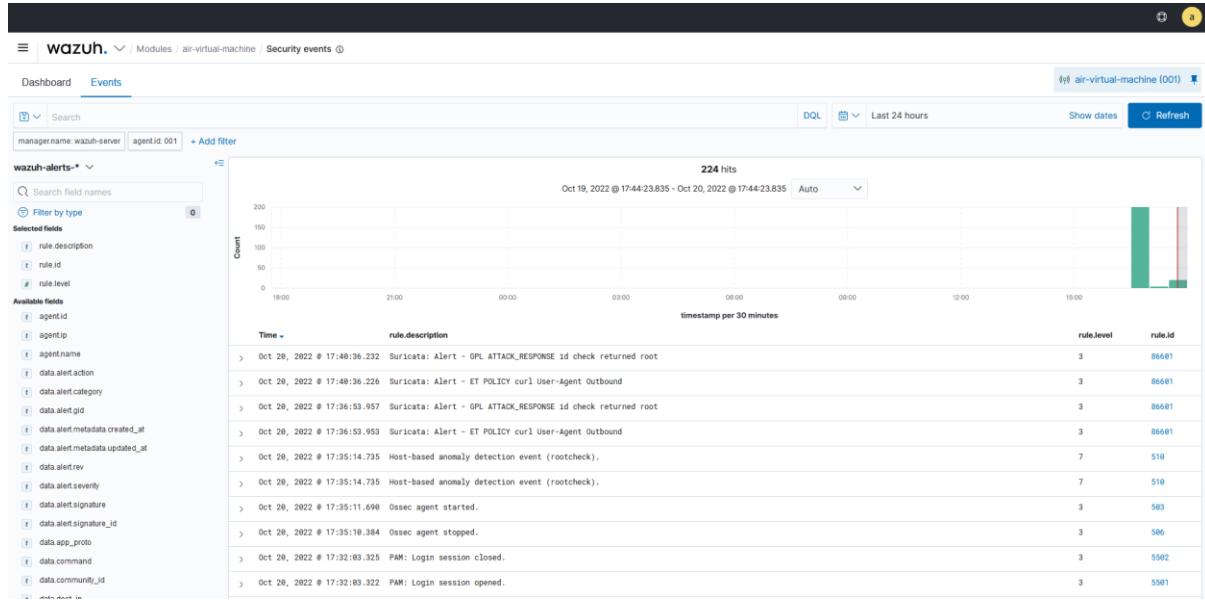
**Step 2:** Run the following command to capture the logs

“  
`Sudo tail -f /var/log/suricata/eve.json | jq ‘select(.event_type==”alert”)’`”

- And again, now ping the Ubuntu and the logs will be displayed in json format in the log files

```
root@air-virtual-machine:/home/air# sudo tail -f /var/log/suricata/eve.json | jq 'select(.event_type=="alert")'
{
  "timestamp": "2022-10-20T15:55:23.913565+0530",
  "flow_id": 1862233966768285,
  "in_iface": "ens33",
  "event_type": "alert",
  "src_ip": "192.168.152.1",
  "src_port": 0,
  "dest_ip": "192.168.152.129",
  "dest_port": 0,
  "proto": "ICMP",
  "icmp_type": 8,
  "icmp_code": 0,
  "community_id": "1:sJBDU0FD/jIhWT2WX23zmszoRHm=",
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 1,
    "rev": 1,
    "signature": "ICMP Ping",
    "category": "",
    "severity": 3
  }
},
```

- And the suricata data will be pushed into the wazuh-server where the logs will be displayed in the json format for clear understanding.



## Installation of ZEEK

### **WHAT IS ZEEK:**

Zeek is a passive, open-source network traffic analyser tool. Many companies and operators use Zeek as NSM (Network Security Monitor) which is very helpful for investigating suspicious or malicious activity in the network. Zeek also supports a wide range of traffic analysis tasks beyond the cyber security domain, including performance measurement and troubleshooting too.

Zeek logs the network activity in a separate file. These logs include all the data like HTTP sessions, requested URIs, MIME types, server responses, DNS requests, SSL certificates, Key content of SMTP sessions and much more. All the data is written in a well-structured tab separated or JSON format

In addition to this, Zeek can also perform few special tasks like, extracting files from HTTP sessions, detecting malware, detects SSH brute forcing, validating SSL certs.

**Step 1:** Open the Ubuntu terminal and Add Zeek repository to sources.list file

Command:“

echo'deb

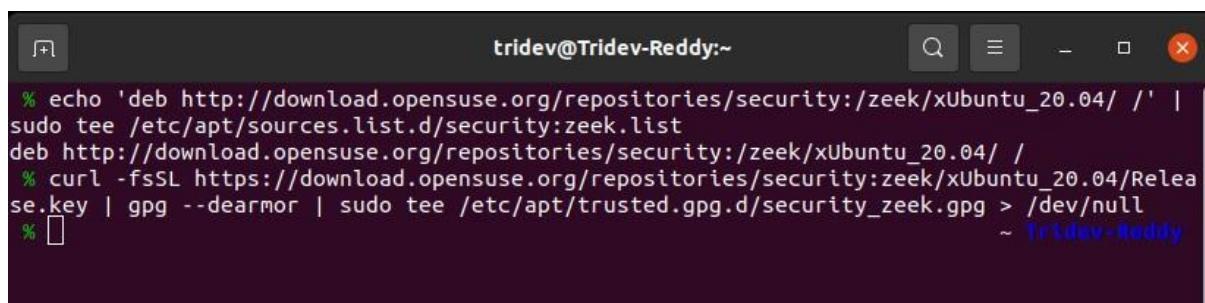
[http://download.opensuse.org/repositories/security:/zeek/xUbuntu\\_22.0](http://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.0)  
4//sudo tee/etc/apt/sources.list.d/security:zeek.list “

After running the above command ,execute the following command:

Command: “ curl -fsSL

[https://download.opensuse.org/repositories/security:zeek/xUbuntu\\_20.04/Release.key](https://download.opensuse.org/repositories/security:zeek/xUbuntu_20.04/Release.key)  
| gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/security\_zeek.gpg > /dev/null “

The output looks like:



```
tridev@Tridev-Reddy:~ % echo 'deb http://download.opensuse.org/repositories/security:/zeek/xUbuntu_20.04/ /' | sudo tee /etc/apt/sources.list.d/security:zeek.list
deb http://download.opensuse.org/repositories/security:/zeek/xUbuntu_20.04/ /
% curl -fsSL https://download.opensuse.org/repositories/security:zeek/xUbuntu_20.04/Release.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/security_zeek.gpg > /dev/null
% 
```

**Step 2:** Update the system repositories with following command “ sudo apt update “

**Step 3:** Install Zeek from apt “ sudo apt install zeek “

- During the installation, you will be asked to select some Postfix settings, select Internet Site and remaining leave default

```
% sudo apt install zeek
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libbroker-dev libmaxminddb-dev postfix python3-git python3-gitdb
  python3-semantic-version python3-smmmap zeek-btest zeek-btest-data zeek-core
  zeek-core-dev zeek-libcaf-dev zeek-zkg zeekctl
Suggested packages:
  procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb
  postfix-sqlite sasl2-bin | dovecot-common resolvconf postfix-cdb postfix-doc
  python-git-doc python-semantic-version-doc python3-nose
The following NEW packages will be installed:
  libbroker-dev libmaxminddb-dev postfix python3-git python3-gitdb
  python3-semantic-version python3-smmmap zeek zeek-btest zeek-btest-data zeek-core
  zeek-core-dev zeek-libcaf-dev zeek-zkg zeekctl
0 upgraded, 15 newly installed, 0 to remove and 36 not upgraded.
Need to get 22.4 MB of archives.
After this operation, 89.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

**Step 4:** Add Zeek binary path to PATH

Command:- “ echo "export PATH=\$PATH:/opt/zeek/bin" >> ~/.bashrc  
source ~/.bashrc “

**Step 5:** Now we need to configure the .cfg file, we need to mention the local network to monitor. This is to be mentioned in *opt/zeek/etc/networks.cfg*

The output of the file should look similar to the following figure.

```
GNU nano 4.8          /opt/zeek/etc/networks.cfg
# List of local networks in CIDR notation, optionally followed by a
# descriptive tag.
# For example, "10.0.0.0/8" or "fe80::/64" are valid prefixes.

10.0.0.0/8           Private IP space
172.16.0.0/12         Private IP space
192.168.0.0/16        Private IP space
```

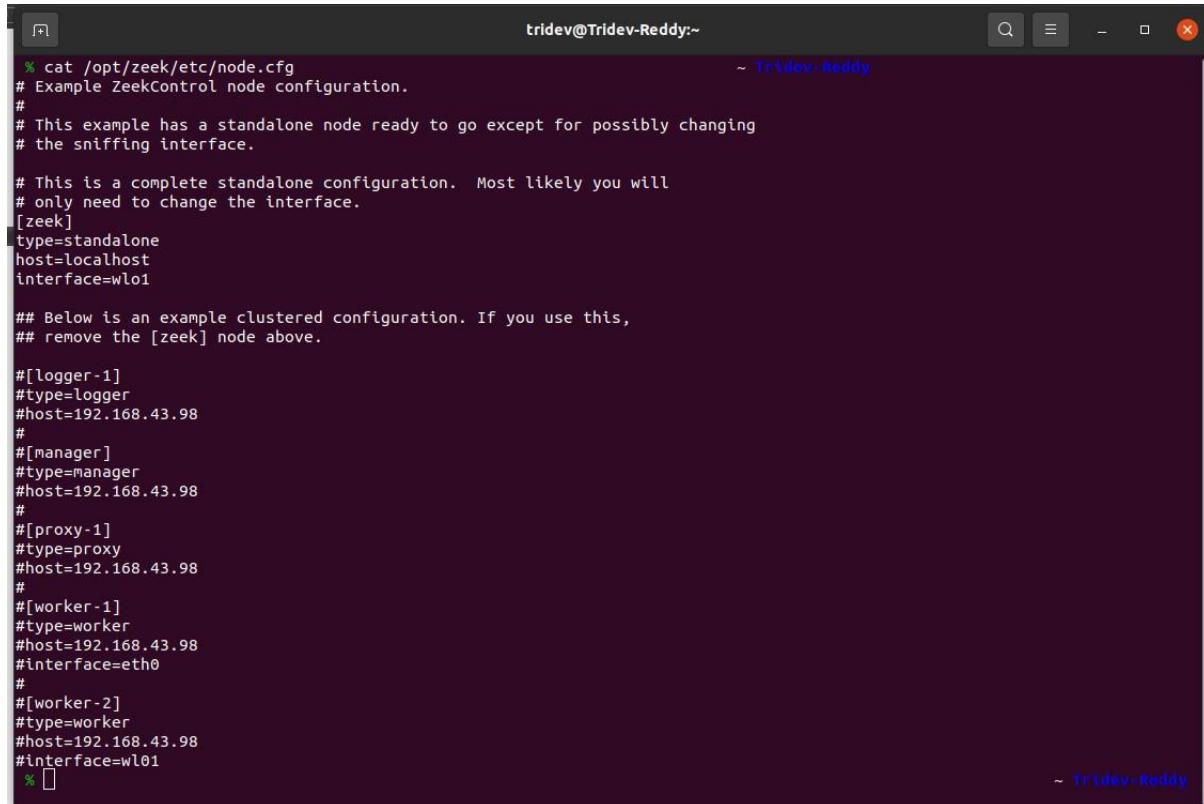
- Zeek can be run in standalone mode or in a cluster setup. To define whether to run in a cluster or standalone setup, you need to edit the `/opt/zeek/etc/node.cfg` configuration file.

For a standalone configuration, there must be only one Zeek node defined in this file.

For a cluster configuration, at a minimum there must be a manager node, a proxy node, and one or more worker nodes.

- As we are using standalone now, just leave everything default and change the interface name in [zeek] to the interface of your network.

The output should look like the following figure



```

tridev@Tridev-Reddy:~ % cat /opt/zeek/etc/node.cfg
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# the sniffing interface.

# This is a complete standalone configuration. Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=wlo1

## Below is an example clustered configuration. If you use this,
## remove the [zeek] node above.

#[logger-1]
#type=logger
#host=192.168.43.98
#
#[manager]
#type=manager
#host=192.168.43.98
#
#[proxy-1]
#type=proxy
#host=192.168.43.98
#
#[worker-1]
#type=worker
#host=192.168.43.98
#interface=eth0
#
#[worker-2]
#type=worker
#host=192.168.43.98
#interface=wlo1
% 

```

**Step 5:** Now we need to validate the configurations. For that execute the following command

`cd opt/zeek/bin`

`sudo ./zeekctl check`

Once the settings are configured properly, the output should look like



```

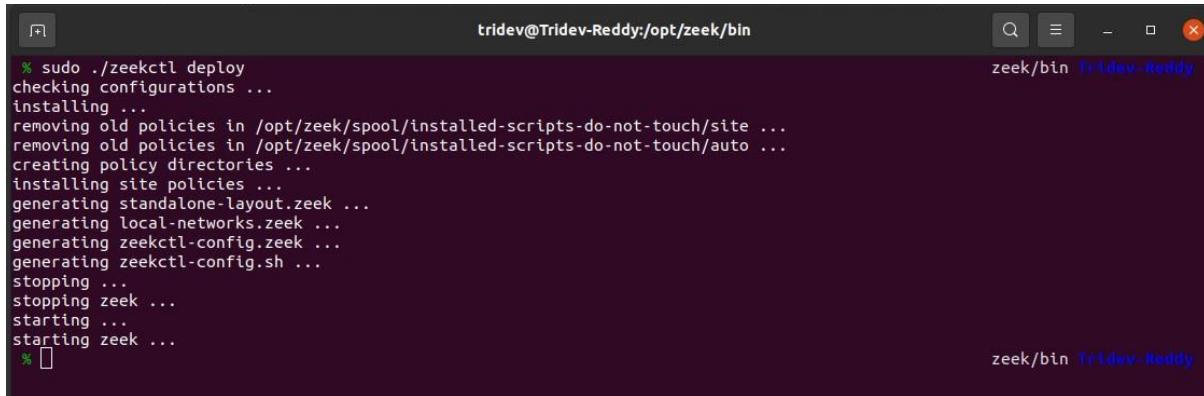
tridev@Tridev-Reddy:/opt/zeek/bin % sudo ./zeekctl check
zeek scripts are ok.
% 

```

## Step 6: Checking zeekctl configuration

- Now we need to deploy zeek configurations. For doing that execute the following command in the same directory

**Command: “ sudo ./zeekctl deploy ”**



```
tridev@Tridev-Reddy:~/opt/zeek/bin
% sudo ./zeekctl deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
starting ...
starting zeek ...
%
zeek/bin Tridev-Reddy
```

## Step 7: Now to check the Instance whether the Zeek is running or not, execute the following command “ sudo ./zeekctl status ”



```
tridev@Tridev-Reddy:~/opt/zeek/bin
% sudo ./zeekctl status
Name      Type      Host      Status      Pid      Started
zeek      standalone localhost  running    9997  26 Jan 14:07:19
%
zeek/bin Tridev-Reddy
```

## DEMO:

Now let's have a quick demo on how to sniff data and analyse logs using zeek.  
Start zeek using the command

**./zeekctl start**

For generating some traffic, I am crawled through the webpage and grabbed index.html from google.com and also pinged google. (ping 8.8.8.8)

```

root@Tridev-Reddy:/opt/zeek/bin# ls /opt/zeek/logs/current
loaded_scripts.log  packet_filter.log  stats.log  stderr.log  stdout.log
root@Tridev-Reddy:/opt/zeek/bin# wget https://google.com/
--2022-01-26 13:51:53-- https://google.com/
Resolving google.com (google.com)... 142.250.77.78, 2404:6800:4009:829::200e
Connecting to google.com (google.com)|142.250.77.78|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.google.com/ [following]
--2022-01-26 13:51:54-- https://www.google.com/
Resolving www.google.com (www.google.com)... 172.217.166.36, 2404:6800:4009:813::2004
Connecting to www.google.com (www.google.com)|172.217.166.36|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html.1'

index.html.1 [ => ] 17.70K 64.4KB/s in 0.3s

2022-01-26 13:51:56 (64.4 KB/s) - 'index.html.1' saved [18120]

root@Tridev-Reddy:/opt/zeek/bin# ls /opt/zeek/logs/current
capture_loss.log  known_hosts.log  packet_filter.log  stderr.log
conn.log          known_services.log  ssl.log        stdout.log
dns.log          loaded_scripts.log  stats.log

```

- There are few more file in logs/current directory. Once the zeek is stopped, they will be stored in a separate new folder with today's date. After stopping the zeek tool, the files in current directory will be deleted, no worries they will be stored in a new folder safely

Let's see a file named conn.log and analyse the data.

```

root@Tridev-Reddy:/opt/zeek/bin# cat /opt/zeek/logs/current/conn.log
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path conn
#open 2022-01-26-13-51-54
#fields ts      uid      id.orig_h      id.orig_p      id.resp_h      id.resp_p      pr
oto      service duration      orig_bytes      resp_bytes      conn_state      local_orig
local_resp      missed_bytes      history orig_pkts      orig_ip_bytes      resp_pkts      re
sp_ip_bytes      tunnel_parents
#types time      string      addr      port      addr      port      enum      string      interval
count      string      bool      count      string      count      count      count      co
unt      count      string      bool      count      string      count      count      se
t[string]
1643185304.820233      CF9Ly13q7X1gG9GDe1      192.168.43.98      47601      192.168.43.1      53
udp      dns      0.054358      34      73      SF      T      T      0      Dd      16
2      1      101      -
1643185313.647562      C2r2T48xRnb14shDc      192.168.43.98      45106      142.250.77.78      44
3      tcp      ssl      2.333102      653      7932      SF      T      F      0      Sh
ADadtFf 18      1621      16      10501      -
1643185314.924517      CPce5q3rNguUstfd93      192.168.43.98      53078      172.217.166.36      44
3      tcp      ssl      1.261748      637      24309      SF      T      F      0      Sh
ADadtFf 28      2149      26      25669      -
1643185263.450214      CXlWVj1Y437JFiehZ3      192.168.43.98      44518      142.250.77.46      44

```

- From the above figure the complete data that we sniffed. The IP from which the traffic is generated, to which IP, uid, time and all the data. This is how we use zeek to analyse network traffic
- One more cool feature is, we can import any pcap file and analyse the traffic from that file using the argument **-r**

Command: **`./zeek -r`**

So, this is the complete intro and practical demo on how to install and use zeek to analyse network traffic.

### **Faster log analysis with zeek-cut and unix tools**

With high amount of network transactions, the log files become over cluttered and difficult to read. In such scenarios, we may want to filter out only certain columns and/or certain rows. In such cases, the utility **zeekcut** provided as a part of the zeek package itself can be of great help. **Zeek-cut** allows us to pick up certain columns of data.

The syntax of the same is simple

**`./zeek-cut < log_file_name columns_to_be_extracted`**

For example, we may try to pick out only the timestamp, the origin address, the destination address and the protocol used from a certain conn.log file.

This can be done by running

**`./zeek-cut < conn.log ts id.orig_h id.resp_h proto`**

It's still a lot of information. Now Unix tools like **grep** and **sort** can be used to manipulate the data more. For example, from here, we may want to filter out only the **udp** packets. The **grep** command can be used for this.

```
./zeek-cut < conn.log ts id.orig_h id.resp_h proto | grep udp
```

- the output on filtering a particular data **udp** using **grep**.

```
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current
File Edit View Search Terminal Help
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current$ zeek-cut < conn.log ts id.orig_h id.resp_h proto
1643177770.397849 192.168.29.35 224.0.0.251 udp
1643177781.941311 2405:201:800b:a01d:cd85 2405:201:800b:a01d:c0a8:1d01 udp
1643177781.943860 2405:201:800b:a01d:cd85 2405:201:800b:a01d:c0a8:1d01 udp
1643177781.946362 2405:201:800b:a01d:cd85 2405:201:800b:a01d:c0a8:1d01 udp
1643177806.868729 2405:201:800b:a01d:cd85 2404:6800:4009:82c::2004 tcp
1643177807.994564 2405:201:800b:a01d:cd85 2404:6800:4009:82d::2003 tcp
1643177808.089488 2405:201:800b:a01d:cd85 2404:6800:4002:809::200e tcp
1643177808.155352 2405:201:800b:a01d:cd85 2404:6800:4009:801::2001 tcp
1643177808.334963 192.168.29.146 142.250.206.142 tcp
1643177808.083978 2405:201:800b:a01d:cd85 2404:6800:4009:81f::200e tcp
1643177808.497713 2405:201:800b:a01d:cd85 2404:6800:4009:831::2003 tcp
1643177808.559860 2405:201:800b:a01d:cd85 2404:6800:4009:831::2003 tcp
1643177809.234191 2405:201:800b:a01d:cd85 2404:6800:4002:81f::200e tcp
1643177810.245916 2405:201:800b:a01d:cd85 2404:6800:4002:81a::2002 tcp
1643177810.458475 2405:201:800b:a01d:cd85 2404:6800:4009:827::2002 tcp
1643177810.648952 2405:201:800b:a01d:cd85 2404:6800:4002:824::2002 tcp
1643177812.375849 2405:201:800b:a01d:cd85 2404:6800:4002:81f::200e tcp
1643177885.903438 2405:201:800b:a01d:cd85 2404:6800:4009:801::2001 tcp
1643177886.854372 2405:201:800b:a01d:cd85 2405:201:800b:a01d:c0a8:1d01 udp
1643177806.854552 2405:201:800b:a01d:cd85 2405:201:800b:a01d:c0a8:1d01 udp
```

- It still seems the amount of information is too much. We may want to see only **udp** packets, communicating with **224.0.0.251**. This can be done by piping the output from this command to another **grep**.

```
./zeek-cut < conn.log ts id.orig_h id.resp_h proto | grep udp | grep 224.0.0.251
```

```
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current
File Edit View Search Terminal Help
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current$ zeek-cut < conn.log ts id.orig_h id.resp_h proto | grep udp | grep 224.0.0.251
1643177770.397849 192.168.29.35 224.0.0.251 udp
1643177801.730136 192.168.29.128 224.0.0.251 udp
1643177832.656408 192.168.29.128 224.0.0.251 udp
1643177885.901954 192.168.29.35 224.0.0.251 udp
1643177907.815224 192.168.29.35 224.0.0.251 udp
1643177927.886280 192.168.29.35 224.0.0.251 udp
1643177947.750681 192.168.29.35 224.0.0.251 udp
1643177967.821122 192.168.29.35 224.0.0.251 udp
1643177987.891073 192.168.29.35 224.0.0.251 udp
1643178007.757886 192.168.29.35 224.0.0.251 udp
1643178027.826729 192.168.29.35 224.0.0.251 udp
1643178047.898559 192.168.29.35 224.0.0.251 udp
1643178067.762118 192.168.29.35 224.0.0.251 udp
1643178087.832039 192.168.29.35 224.0.0.251 udp
1643178107.901755 192.168.29.35 224.0.0.251 udp
1643178127.767081 192.168.29.35 224.0.0.251 udp
1643178147.836831 192.168.29.35 224.0.0.251 udp
1643178167.908228 192.168.29.35 224.0.0.251 udp
1643178187.772026 192.168.29.35 224.0.0.251 udp
1643178207.843654 192.168.29.35 224.0.0.251 udp
1643178227.912502 192.168.29.35 224.0.0.251 udp
1643178247.777708 192.168.29.35 224.0.0.251 udp
1643178267.847172 192.168.29.35 224.0.0.251 udp
1643178287.917803 192.168.29.35 224.0.0.251 udp
1643178307.782366 192.168.29.35 224.0.0.251 udp
1643178327.852006 192.168.29.35 224.0.0.251 udp
1643178347.757540 192.168.29.35 224.0.0.251 udp
1643178367.787291 192.168.29.35 224.0.0.251 udp
1643178387.857383 192.168.29.35 224.0.0.251 udp
1643178407.927505 192.168.29.35 224.0.0.251 udp
1643178427.792992 192.168.29.35 224.0.0.251 udp
1643178447.862920 192.168.29.35 224.0.0.251 udp
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current$ It still seems the amount of information is too much. We may want to see only udp packets, communicating with 224.0.0.251. This can be done by piping the output from this command to another grep.
```

- the output when the data is filtered for the particular address.

- Now, we may want to sort the data according to descending order of the time stamp. It can be done by piping this output to **sort** command.
- **./zeek-cut < conn.log ts id.orig\_h id.resp\_h proto | grep udp | grep 224.0.0.251 | sort -n -r**

```
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current$ ./zeek-cut < conn.log ts id.orig_h id.resp_h proto | grep udp | grep 224.0.0.251 | sort -n -r
File Edit View Search Terminal Help
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current$ zeek-cut < conn.log ts id.orig_h id.resp_h proto | grep udp | grep 224.0.0.251 | sort -n -r
1643178447.862920 192.168.29.35 224.0.0.251 udp
1643178427.792992 192.168.29.35 224.0.0.251 udp
1643178407.927505 192.168.29.35 224.0.0.251 udp
1643178387.857383 192.168.29.35 224.0.0.251 udp
1643178367.787291 192.168.29.35 224.0.0.251 udp
1643178347.757540 192.168.29.35 224.0.0.251 udp
1643178327.852806 192.168.29.35 224.0.0.251 udp
1643178307.782366 192.168.29.35 224.0.0.251 udp
1643178287.917803 192.168.29.35 224.0.0.251 udp
1643178267.847172 192.168.29.35 224.0.0.251 udp
1643178247.777768 192.168.29.35 224.0.0.251 udp
1643178227.912502 192.168.29.35 224.0.0.251 udp
1643178207.843654 192.168.29.35 224.0.0.251 udp
1643178187.772926 192.168.29.35 224.0.0.251 udp
1643178167.968228 192.168.29.35 224.0.0.251 udp
1643178147.836831 192.168.29.35 224.0.0.251 udp
1643178127.767081 192.168.29.35 224.0.0.251 udp
1643178107.961755 192.168.29.35 224.0.0.251 udp
1643178087.832039 192.168.29.35 224.0.0.251 udp
1643178067.762118 192.168.29.35 224.0.0.251 udp
1643178047.898559 192.168.29.35 224.0.0.251 udp
1643178027.826729 192.168.29.35 224.0.0.251 udp
1643178007.757886 192.168.29.35 224.0.0.251 udp
1643177987.891973 192.168.29.35 224.0.0.251 udp
1643177967.821122 192.168.29.35 224.0.0.251 udp
1643177947.750681 192.168.29.35 224.0.0.251 udp
1643177927.886786 192.168.29.35 224.0.0.251 udp
1643177907.815224 192.168.29.35 224.0.0.251 udp
1643177885.981954 192.168.29.35 224.0.0.251 udp
1643177832.656408 192.168.29.128 224.0.0.251 udp
1643177801.730136 192.168.29.128 224.0.0.251 udp
1643177770.397849 192.168.29.35 224.0.0.251 udp
procoder101@procoder101-Inspiron-3593:/usr/local/zeek/logs/current$
```

## **ZEEK INTEGRATION WITH ELK STACK**

In this procedure we will use a tool called Filebeat, which monitors, collects and forwards the logs to the Elastic Search. We will configure Filebeat with Zeek, so that the data collected by zeek, will be forwarded and centralized in our Kibana Dashboard

### **INSTALLING FILEBEAT**

Now it's time to setup Filebeat with our Zeek. Follow the steps one by one to install and configure filebeat.

**Step 1:** Install Filebeat using apt

**Command:** - “ sudo apt install filebeat ”

**Step 2:** Now we need to configure the .yml file which is present in etc/filebeat/ folder “ sudo nano /etc/filebeat/filebeat.yml ”

We need to configure only two things here. In the Filebeat Input section, change the type to log and uncomment the enabled: false and change it to true. Including that, we need to specify the path of where the logs are stored, i.e we need to specify /opt/zeek/logs/current/\*.log

Once it is done the first part of settings should look similar to the following figure.

```
# Configuration file.
# ===== Filebeat inputs =====
filebeat.inputs:
  # Each - is an input. Most options can be set at the input level, so
  # you can use different inputs for various configurations.
  # Below are the input specific configurations.

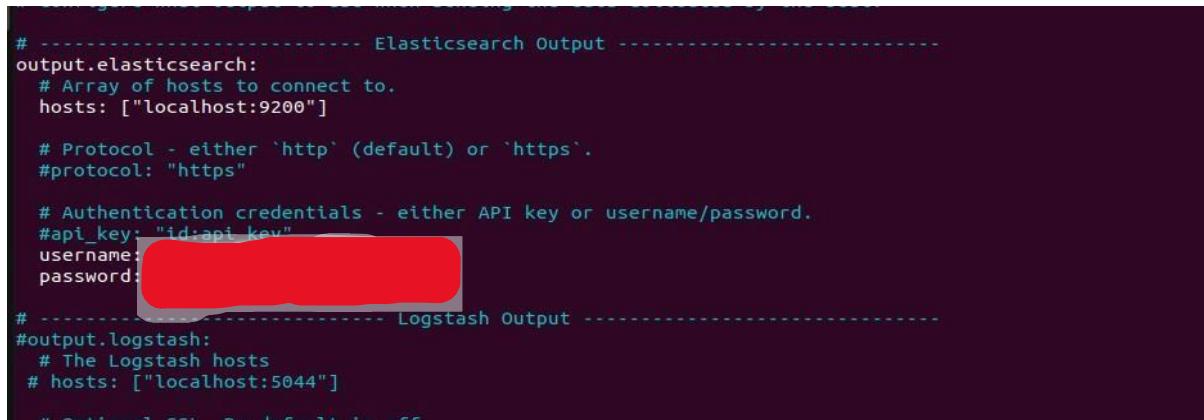
  # filestream is an input for collecting log messages from files.
  - type: log

    # Change to true to enable this input configuration.
    enabled: true

    # Paths that should be crawled and fetched. Glob based paths.
    paths:
      - /var/log/*.log
      - /opt/zeek/logs/current/*.log
      #- c:\programdata\elasticsearch\logs\*

    # Exclude lines. A list of regular expressions to match. It drops the lines that are
    # matching any regular expression from the list.
    #exclude_lines: ['^DBG']
```

- The second thing to be changed in Elasticsearch output section under Outputs. Uncomment the output.elasticsearch and hosts. Makesure the url of host and port number is similar to one you configured while installing ELK. I kept it as localhost with port number 9200.
- In the same section, below uncomment the username and password, and enter the username and password of elastic user that you generated while configuring ELK after installation.



```

# ----- Elasticsearch Output -----
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api-key"
  username: [REDACTED]
  password: [REDACTED]

# ----- Logstash Output -----
#output.logstash:
  # The Logstash hosts
  # hosts: ["localhost:5044"]

  # Optional SSL. By default is off.

```

**Step 3:** Now we need to enable the Zeek module in Filebeat so that it forwards the logs from Zeek. Execute the following command.

“ sudo filebeat modules enable zeek ”

- We are almost ready, the last step is to that we need to configure zeek.yml file to mention what type of data to be logged. This can be done by modifying /etc/filebeat/modules.d/zeek.yml file.

In that .yml file, we need to mention the directory where that specified logs are stored. We know that the logs are stored in current folder. In that we have several files like dns.log, conn.log, dhcp.log and many more. We need to mention each path in respective section. You can leave unwanted by changing the enabled value to false if and only if you don't want logs from that file/program.

For example, for dns, makesure the enabled value is true and path to be mentioned as:

var.paths: [ "/opt/zeek/logs/current/dns.log", "/opt/zeek/logs/\*.dns.json" ]

- Similarly for remaining all. I did for few and that I needed. I added everything mainly required, you can keep everything the same that I mentioned.

```
curl -H "Content-Type: application/json" -XPUT "http://localhost:5601/_modules/zeek/_all/_source" -d @zeek.yml
```

```
# Module: zeek
# Docs: https://www.elastic.co/guide/en/beats/filebeat/7.16/filebeat-module-zeek.html

- module: zeek
  capture_loss:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/capture_loss.log", "/opt/zeek/logs/*.capture_loss.json" ]
  connection:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/conn.log", "/opt/zeek/logs/*.conn.json" ]
  dce_rpc:
    enabled: false
  dhcp:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/dhcp.log", "/opt/zeek/logs/*.dhcp.json" ]
  dnsp:
    enabled: false
  dns:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/dns.log", "/opt/zeek/logs/*.dns.json" ]
  dpd:
    enabled: false
  files:
    enabled: false
  ftp:
    enabled: false
  http:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/http.log", "/opt/zeek/logs/*.https.json" ]
  intel:
    enabled: false
  irc:
    enabled: false
```

- Everything was done, it's time to start the filebeat. Execute the following commands  
**“ sudo filebeat setup ”**  
**“ sudo service filebeat start ”**

Everything is completed, let's move to our Kibana dashboard and check whether we are receiving the data from Zeek via Filebeat or not.

## **Installing the Elasticsearch in Wazuh-manager**

- Elasticsearch is a distributed, free and open search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.

### **How does Elasticsearch work?**

Raw data flows into Elasticsearch from a variety of sources, including logs, system metrics, and web applications. *Data ingestion* is the process by which this raw data is parsed, normalized, and enriched before it is *indexed* in Elasticsearch. Once indexed in Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data. From Kibana, users can create powerful visualizations of their data, share dashboards, and manage the Elastic Stack.

### **What is Elasticsearch used for?**

The speed and scalability of Elasticsearch and its ability to index many types of content mean that it can be used for a number of use cases:

- Application search
- Website search
- Enterprise search
- Logging and log analytics
- Infrastructure metrics and container monitoring
- Application performance monitoring
- Geospatial data analysis and visualization
- Security analytics
- Business analytics

### **Elasticsearch cluster**

The Elastic Stack can be installed as a single-node cluster or as a multi-node cluster. The single-node installation will be performed in only one host where Open Distro for Elasticsearch will be installed. The multi-node installation consists of the installation of several Elastic Stack nodes in different hosts that will communicate between them. This kind of installation provides high availability and load balancing.

## Installing Elasticsearch single-node cluster

### Installing prerequisites

Some extra packages are needed for the installation, such as `curl` or `unzip`, that will be used in further steps:

**Step 1:** Install all the necessary packages:

- `yum install zip unzip curl`
- Adding the Elastic Stack repository

**Step 2:** Import the GPG key:

- `rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch`

**Step 3:** Add the repository by running the following command

“`cat > /etc/yum.repos.d/elastic.repo << EOF`”

```
- [elasticsearch-7.x]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
EOF
```

## Elasticsearch installation and configuration

**Step 1:** Install the Elasticsearch package:

- yum install elasticsearch-7.17.6

**Step 2:** Once Elasticsearch is installed, it can be configured by downloading the file [/etc/elasticsearch/elasticsearch.yml](#):

```
# curl -so /etc/elasticsearch/elasticsearch.yml  
https://packages.wazuh.com/4.3/tpl/elastic-basic/elasticsearch.yml
```

### Certificates creation and deployment

The number of Wazuh servers to be implemented will determine the next step. Select Wazuh single-node cluster, if there is only one Wazuh server, or Wazuh multi-node cluster in case there are two or more Wazuh servers.

**Step 1:** The instances file can be created /usr/share/elasticsearch/instances.yml as follows:

```
cat > /usr/share/elasticsearch/instances.yml <<\EOF
```

```
instances:  
- name: "elasticsearch"  
  ip:  
  - "10.0.0.2"  
- name: "filebeat"  
  ip:  
  - "10.0.0.3"  
- name: "kibana"  
  ip:  
  - "10.0.0.4"  
EOF
```

- Every name section corresponds to one host in the Wazuh Server - Elastic Stack environment. In this example, the file describes:
  - An elasticsearch instance with IP address 10.0.0.2.
  - A filebeat instance with IP address 10.0.0.3 corresponding to a single-node Wazuh cluster.
  - A kibana instance with IP address 10.0.0.4. If Kibana will be installed in the same server as Elasticsearch, the same IP address may be used.

Replace the IPs with the corresponding addresses for each instance in your environment.

**Step 3:** Create the certificates using the elasticsearch-certutil tool by using following command:

```
- /usr/share/elasticsearch/bin/elasticsearch-certutil cert ca --pem --in instances.yml --keep-ca-key --out ~/certs.zip
```

**Step 4:** Copy ~/certs.zip to all the servers of the distributed deployment. This can be done by using, for example, scp.

**Step 5:** The next step is to create the directory `/etc/elasticsearch/certs`, and then copy the certificate authorities, the certificate and key there

```
unzip ~/certs.zip -d ~/certs
mkdir /etc/elasticsearch/certs/ca -p
cp -R ~/certs/ca/ ~/certs/elasticsearch/* /etc/elasticsearch/certs/
chown -R elasticsearch: /etc/elasticsearch/certs
chmod -R 500 /etc/elasticsearch/certs
chmod 400 /etc/elasticsearch/certs/ca/ca.* /etc/elasticsearch/certs/elasticsearch.*
rm -rf ~/certs/
```

**Step 6:** If you are going to install Kibana in this node, keep the certificates file. Otherwise, if the file has been copied already to all the instances of the distributed deployment, remove it to increase security `rm -f ~/certs.zip`.

**Step 7:** Enable and start the Elasticsearch service:

```
systemctl daemon-reload
systemctl enable elasticsearch
systemctl start elasticsearch
```

**Step 8:** Generate credentials for all the Elastic Stack pre-built roles and users:

```
- /usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
```

The command above will prompt an output like this. Save the password of the elastic user for further steps:

**OUTPUT:**

```
Changed password for user apm_system
PASSWORD apm_system = ILPZhZkB6oUOzzCrkLSF
```

```
Changed password for user kibana_system
PASSWORD kibana_system = TaLqVOnSoqKTYLIU0vDn
```

Changed password for user kibana  
PASSWORD kibana = TaLqVOvXoqKTYLIU0vDn

Changed password for user logstash\_system  
PASSWORD logstash\_system = UtuDv2tWkXGYL83v9kWA

Changed password for user beats\_system  
PASSWORD beats\_system = qZcbvCslafMpoEOrE9Ob

Changed password for user remote\_monitoring\_user  
PASSWORD remote\_monitoring\_user = LzJpQiSylncmCU2GLBTS

Changed password for user elastic  
PASSWORD elastic = AN4UeQGA7HG15iHpMla7

## **Installing the Wazuh cluster**

- A Wazuh cluster is a group of Wazuh managers that work together to enhance the availability and scalability of the service. With a Wazuh cluster setup, we have the potential to greatly increase the number of agents as long as we add worker nodes whenever necessary.

**Step 1:** Installing the prerequisites and necessary packages

-yum install zip unzip curl

### **Installing Wazuh server**

The Wazuh server collects and analyzes data from deployed agents. It runs the Wazuh manager, the Wazuh API and Filebeat. The first step in setting up Wazuh is adding Wazuh repository to the server. Alternatively, the Wazuh manager package can be downloaded directly, and compatible versions can be checked here.

Step 2: Import the GPG key:

-rpm --import <https://packages.wazuh.com/key/GPG-KEY-WAZUH>

Step 3: Add the repository in the following command

-cat > /etc/yum.repos.d/wazuh.repo << EOF

```
[wazuh]
gpgcheck=1
gpgkey=https://packages.wazuh.com/key/GPG-KEY-WAZUH
enabled=1
name=EL-$releasever - Wazuh
baseurl=https://packages.wazuh.com/4.x/yum/
protect=1
EOF
```

## Installing Kibana

- Kibana is a data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers powerful and easy-to-use features such as histograms, line graphs, pie charts, heat maps, and built-in geospatial support.
- Some extra packages are needed for the installation, such as curl or unzip, that will be used in further steps:

**Step 1:** Install all the necessary packages:

- **yum install zip unzip curl**

➤ **Adding the Elastic Stack repository**

**Step 1:** Import the GPG key:

- **rpm --import <https://artifacts.elastic.co/GPG-KEY-elasticsearch>**

**Step 2:** Add the repository in the following command

“**cat > /etc/yum.repos.d/elastic.repo << EOF**”

```
[elasticsearch-7.x]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
EOF
```

## Kibana installation and configuration

### **Step 1:** Install the Kibana package

- **yum install kibana-7.17.6**

**Step 2:** The next step is the certificate placement, this guide assumes that a copy of certs.zip is placed in the root home folder (~/):

```
unzip ~/certs.zip -d ~/certs  
  
rm -f ~/certs/ca/ca.key  
  
mkdir /etc/kibana/certs/ca -p  
  
cp ~/certs/ca/ca.crt /etc/kibana/certs/ca  
  
cp ~/certs/kibana/* /etc/kibana/certs/  
  
chown -R kibana: /etc/kibana/certs  
  
chmod -R 500 /etc/kibana/certs  
  
chmod 400 /etc/kibana/certs/ca/ca.* /etc/kibana/certs/kibana.*  
  
rm -rf ~/certs ~/certs.zip
```

### **Step 3:** Download the Kibana configuration file:

- **curl -so /etc/kibana/kibana.yml**  
<https://packages.wazuh.com/4.3/tpl/elastic-basic/kibana.yml>

### **Step 4:** Edit the **/etc/kibana/kibana.yml** file:

```
server.host: <kibana_ip>  
elasticsearch.hosts: "https://<elasticsearch_DN>:9200"  
elasticsearch.password: <elasticsearch_password>
```

➤ Values to be replaced:

- <kibana\_ip>: by default, Kibana only listens on the loopback interface (localhost), which means that it can be only accessed from the same machine. To access Kibana from the outside, it may be configured to listen on its network IP address by replacing kibana\_ip with Kibana host IP address.

- <elasticsearch\_DN>: the host's domain name. In case of having more than one Elasticsearch node, Kibana can be configured to connect to multiple Elasticsearch nodes in the same cluster. The nodes' domain names can be separated with commas. Eg. ["https://elasticsearch\_DN1:9200", "https://elasticsearch\_DN2:9200", "https://elasticsearch\_DN3:9200"]
- <elasticsearch\_password>: the password generated during the Elasticsearch installation and configuration for the elastic user.

**Step 5:** Create the `/usr/share/kibana/data` directory:

```
mkdir /usr/share/kibana/data
chown -R kibana:kibana /usr/share/kibana
```

**Step 6:** Install the Wazuh Kibana plugin:

The installation of the plugin must be done from the Kibana home directory.

```
cd /usr/share/kibana
sudo -u kibana /usr/share/kibana/bin/kibana-plugin install
https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.3.10_7.17.6-1.zip
```

**Step 7:** Link Kibana's socket to privileged port 443:

- `setcap 'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node`

**Step 8:** Enable and start the Kibana service:

```
systemctl daemon-reload
systemctl enable kibana
systemctl start kibana
```

**Step 8:** Access the web interface using the password generated during the Elasticsearch installation process:

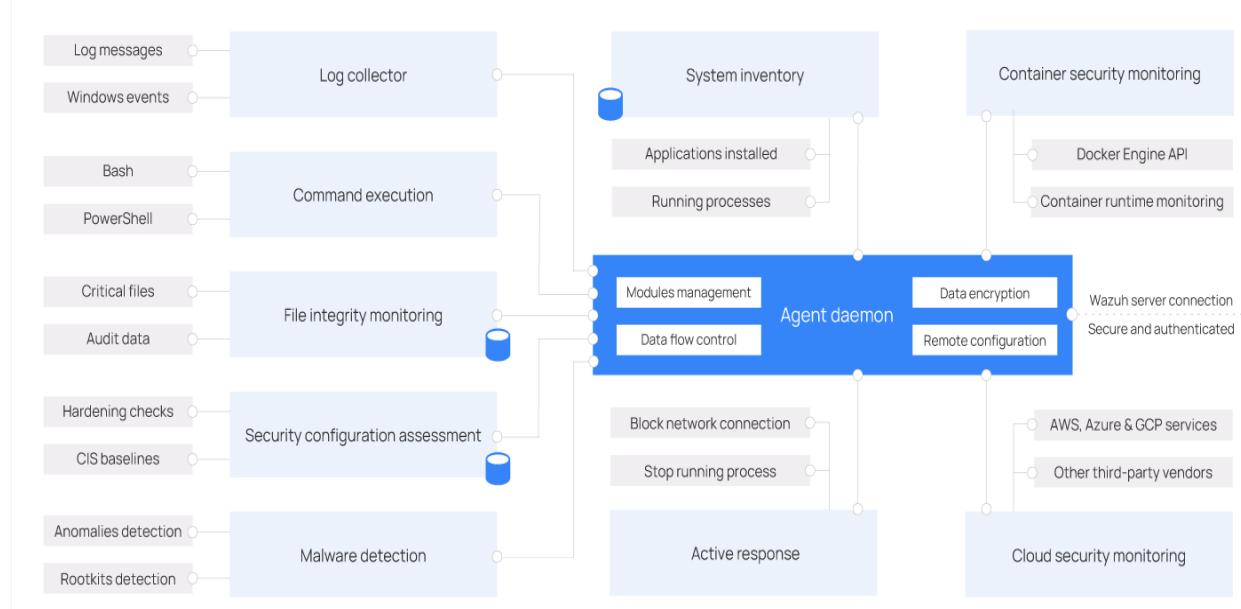
```
URL: https://<kibana_ip>
user: elastic
password: <PASSWORD_elastic>
```

Upon the first access to Kibana, the browser shows a warning message stating that the certificate was not issued by a trusted authority. An exception can be added in the advanced options of the web browser or, for increased security, the root-ca.pem file previously generated can be imported to the certificate manager of the browser. Alternatively, a certificate from a trusted authority can be configured.

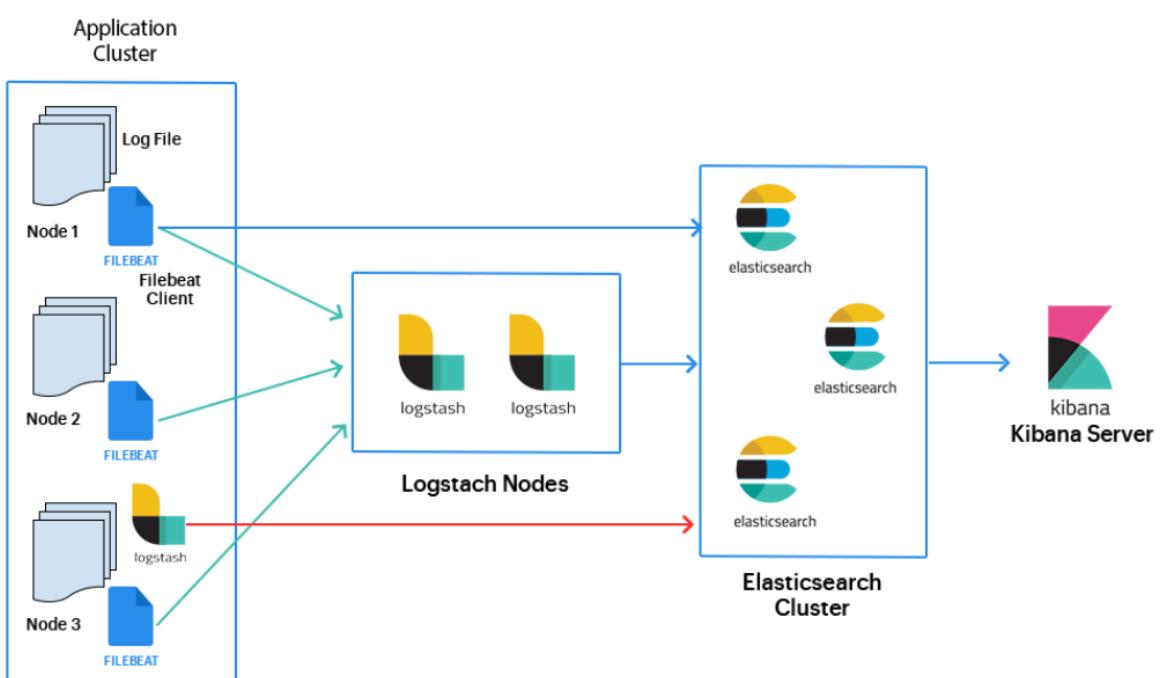
With the first access attempt, the Wazuh Kibana plugin may prompt a message that indicates that it cannot communicate with the Wazuh API. To solve this issue, edit the file `/usr/share/kibana/data/wazuh/config/wazuh.yml` and replace the url with the Wazuh server's address:

```
hosts:  
  - default:  
    url: https://localhost  
    port: 55000  
    username: wazuh-wui  
    password: wazuh-wui  
    run_as: false
```

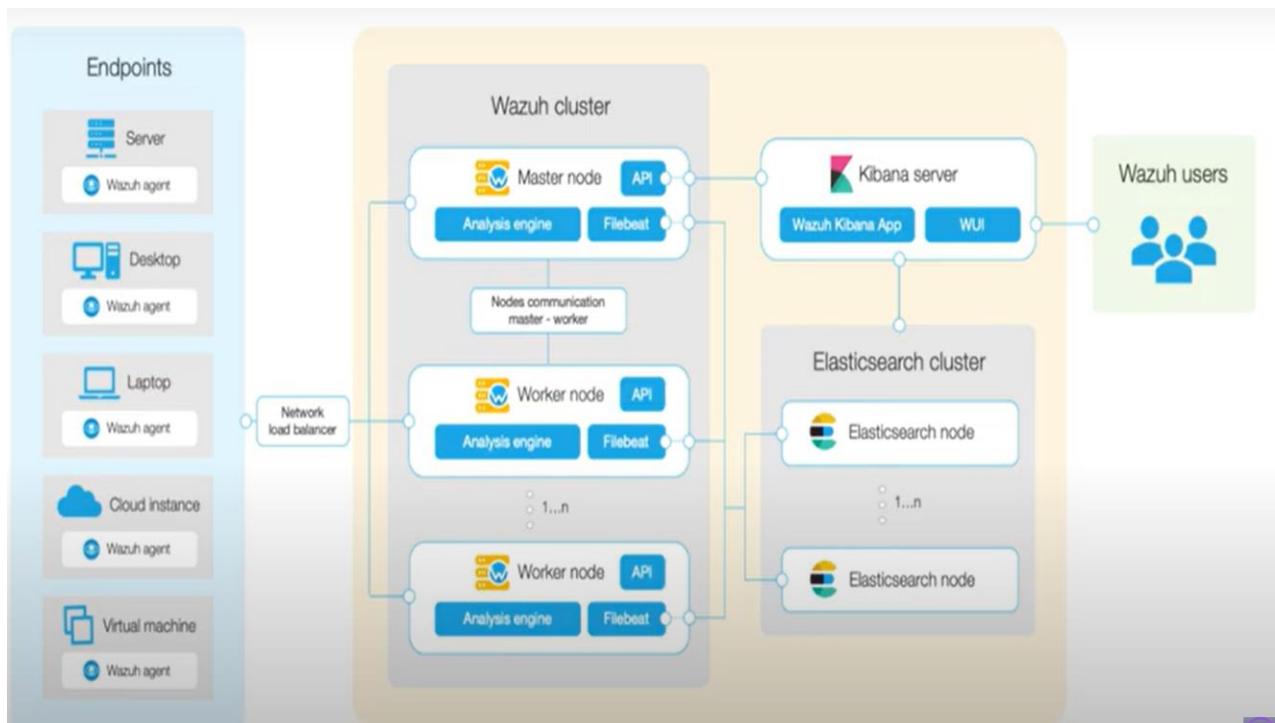
## Wazuh-Agent architecture:



## Kibana flow Diagram:



## Wazuh data flow:



## Wazuh-dashboard:

The dashboard interface includes the following sections:

- Top navigation:** wazuh. / Agents / Debian
- Filtering:** Index pattern: wazuh-alerts-\*; API: env-1
- Table view:** Shows a single row for Agent ID 001 with status active, IP 10.0.1.85, Version Wazuh v4.3.0, Groups default, Operating system Debian GNU/Linux 9, Cluster node master, Registration date Feb 14, 2022 @ 18:03:05.000, and Last keep alive Feb 17, 2022 @ 11:13:13.000.
- MITRE:** Top Tactics: Credential Access (1894), Lateral Movement (27), Impact (2), Initial Access (1).
- Compliance:** PCI DSS chart showing counts for various controls: 10.2.4 (8008), 10.2.5 (8008), 10.6.1 (1842), 11.4 (63), 11.5 (2).
- FIM: Recent events:** A table listing recent integrity check changes for /etc/resolv.conf on Feb 17, 2022, and Feb 16, 2022.
- Events count evolution:** A line graph showing the count of events over time (timestamp per 30 minutes) from 12:00 to 09:00.
- SCA: Last scan:** CIS Benchmark for Debian/Linux 9 (cis\_debian). It shows a score of 38% with 64 Passes, 104 Fails, and 175 Total checks. Scan details: Start time: Feb 17, 2022 @ 06:51:36.000, Duration: < 1s.

## **Acknowledgement:**

The authors are grateful to Sibi Chakkavarthy Sethuraman at the School of Computer Science and Engineering, VIT-AP for their continuous guidance and support. A special thanks to Center for Excellence in Artificial Intelligence and Robotics (AIR).

## **Reference:**

- [1] B. O. Omoyiola, “An overview of root causes of cybersecurity breaches in organizations,” Available at SSRN 4348319, 2023.
- [2] A. Ghorbel, M. Ghorbel, and M. Jmaiel, “Privacy in cloud computing environments: a survey and research challenges,” *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2763–2800, 2017.
- [3] P. Radoglou-Grammatikis, P. Sarigiannidis, E. Iturbe, E. Rios, S. Martinez, A. Sarigiannidis, G. Eftathopoulos, Y. Spyridis, A. Sesis, N. Vakakis et al., “Spear siem: A security information and event management system for the smart grid,” *Computer Networks*, vol. 193, p. 108008, 2021.
- [4] M. M. Nair and A. K. Tyagi, “Privacy: History, statistics, policy, laws, preservation and threat analysis.” *Journal of Information Assurance & Security*, vol. 16, no. 1, 2021.
- [5] A. Kott and I. Linkov, “To improve cyber resilience, measure it,” arXiv preprint arXiv:2102.09455, 2021.
- [6] I. Linkov, D. A. Eisenberg, K. Plourde, T. P. Seager, J. Allen, and A. Kott, “Resilience metrics for cyber systems,” *Environment Systems and Decisions*, vol. 33, pp. 471–476, 2013.
- [7] S. Bagheri and G. Ridley, “Organisational cyber resilience: research opportunities,” in *ACIS2017: Australasian Conference on Information Systems*, 2017, pp. 1–10.
- [8] D. O’Sullivan and L. Dooley, *Applying innovation*. Sage publications, 2008.
- [9] A. Annarelli, F. Nonino, and G. Palombi, “Understanding the management of cyber resilient systems,” *Computers & industrial engineering*, vol. 149, p. 106829, 2020.
- [10] N. Novaes Neto, S. Madnick, M. G. de Paula, N. Malara Borges et al., “A case study of the capital one data breach,” Stuart E. and Moraes G. de Paula, Anchises and Malara Borges, Natasha, *A Case Study of the Capital One Data Breach (January 1, 2020)*, 2020.

- [11] B. Aldous, “Data privacy,” 2022.
- [12] Z. Mohammed, “Data breach recovery areas: an exploration of organization’s recovery strategies for surviving data breaches,” *Organizational Cybersecurity Journal: Practice, Process and People*, vol. 2, no. 1, pp. 41–59, 2022.
- [13] S. Mansfield-Devine, “Ibm: Cost of a data breach,” 2022.
- [14] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, “Phishing attacks: A recent comprehensive study and a new anatomy,” *Frontiers in Computer Science*, vol. 3, p. 563060, 2021.
- [15] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4150, 2021.
- [16] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh, “An intrusion detection system for connected vehicles in smart cities,” *Ad Hoc Networks*, vol. 90, p. 101842, 2019.
- [17] S. U. Rehman, K. F. Thang, and N. S. Lai, “Automated pcb identification and defect-detection system (apids),” *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, p. 297, 2019.
- [18] M. A. Aydin, A. H. Zaim, and K. G. Ceylan, “A hybrid intrusion detection system design for computer network security,” *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.
- [19] P. M. Comar, L. Liu, S. Saha, P.-N. Tan, and A. Nucci, “Combining supervised and unsupervised learning for zero-day malware detection,” in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2022–2030.
- [20] D. A. S. Estay, R. Sahay, M. B. Barfod, and C. D. Jensen, “A systematic review of cyber-resilience assessment frameworks,” *Computers & security*, vol. 97, p. 101996, 2020.
- [21] K. Hausken, “Cyber resilience in firms, organizations and societies,” *Internet of Things*, vol. 11, p. 100204, 2020.
- [22] P. Fraga-Lamas and T. M. Fernández-Caramés, “A review on blockchain technologies for an advanced and cyber-resilient automotive industry,” *IEEE access*, vol. 7, pp. 17 578–17 598, 2019.
- [23] R. K. Baggett and B. K. Simpkins, *Homeland security and critical infrastructure protection*. ABC-CLIO, 2018.
- [24] C. Colicchia, A. Creazza, C. Noè, and F. Strozzi, “Information sharing

in supply chains: a review of risks and opportunities using the systematic literature network analysis (slna)," Supply chain management: an international journal, vol. 24, no. 1, pp. 5–21, 2019.

[25] M. Ahsan, K. E. Nygard, R. Gomes, M. M. Chowdhury, N. Rifat, and J. F. Connolly, "Cybersecurity threats and their mitigation approaches using machine learning—a review," Journal of Cybersecurity and Privacy, vol. 2, no. 3, pp. 527–555, 2022.

[26] U. M. Mbanaso, L. Abrahams, and O. Z. Apene, "Conceptual design of a cybersecurity resilience maturity measurement (crmm) framework," The African Journal of Information and Communication, vol. 23, pp. 1–26, 2019.

[27] A. S. Gómez Vidal, "Improvements in ids: adding functionality to wazuh," 2019.

[28] R. A. Sepúlveda Rodríguez, "Analysis of alternatives for a security information and event management tool in a virtualized environment," Computer Science;, 2018.

[29] S. Einy, C. Oz, and Y. D. Navaei, "The anomaly-and signature-based ids for network security using hybrid inference systems," Mathematical Problems in Engineering, vol. 2021, pp. 1–10, 2021.

[30] A. Waleed, A. F. Jamali, and A. Masood, "Which open-source ids? snort, suricata or zeek," Computer Networks, vol. 213, p. 109116, 2022.

[31] A. R. Muhammad, P. Sukarno, and A. A. Wardana, "Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning," Procedia Computer Science, vol. 217, pp. 1406–1415, 2023.

[32] L. Moilanen, "Collecting logs from docker containers," 2020.

[33] F. Mulyadi, L. A. Annam, R. Promya, and C. Charnsripinyo, "Implementing dockerized elastic stack for security information and event management," in 2020-5th International Conference on Information Technology (InCIT). IEEE, 2020, pp. 243–248.

[34] H. T. Thi, N. D. H. Son, P. T. Duv, and V.-H. Pham, "Federated learning-based cyber threat hunting for apt attack detection in sdn-enabled networks," in 2022 21st International Symposium on Communications and Information Technologies (ISCIT). IEEE, 2022, pp. 1–6.

[35] T. A. Welling, "Application security testing," Ph.D. dissertation, 2022.

[36] S. A. Shaikh, H. Chivers, P. Nobles, J. A. Clark, and H. Chen, "Towards scalable intrusion detection," Network Security, vol. 2009, no. 6, pp. 12–16, 2009.

[37] S. Bezzateev, S. Fomicheva, and G. Zhemelov, “Agent-based zero logon vulnerability detection,” in 2021 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). IEEE, 2021, pp. 1–5.

[38] M. A. Hussein and E. K. Hamza, “Secure mechanism applied to big data for iiot by using security event and information management system (siem).”

[39] M. Hasbi, A. R. A. Nurwa, D. F. Priambodo, and W. R. A. Putra, “Infrastructure as code for security automation and

<https://documentation.wazuh.com/current/getting-started/index.html>

<https://discuss.elastic.co/t/unable-to-start-elasticsearch-from-systemctl-in-centos-7/230025>

<https://www.howtoforge.com/suricata-and-zeek-ids-with-elk-on-ubuntu-20-10/>

<https://kifarunix.com/install-zeek-on-ubuntu/>

<https://stackoverflow.com/questions/58656747/elasticsearch-job-for-elasticsearch-service-failed>

<https://documentation.wazuh.com/current/installation-guide/index.html>

