Achieving Cyber Resilience

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

Contexts:

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.

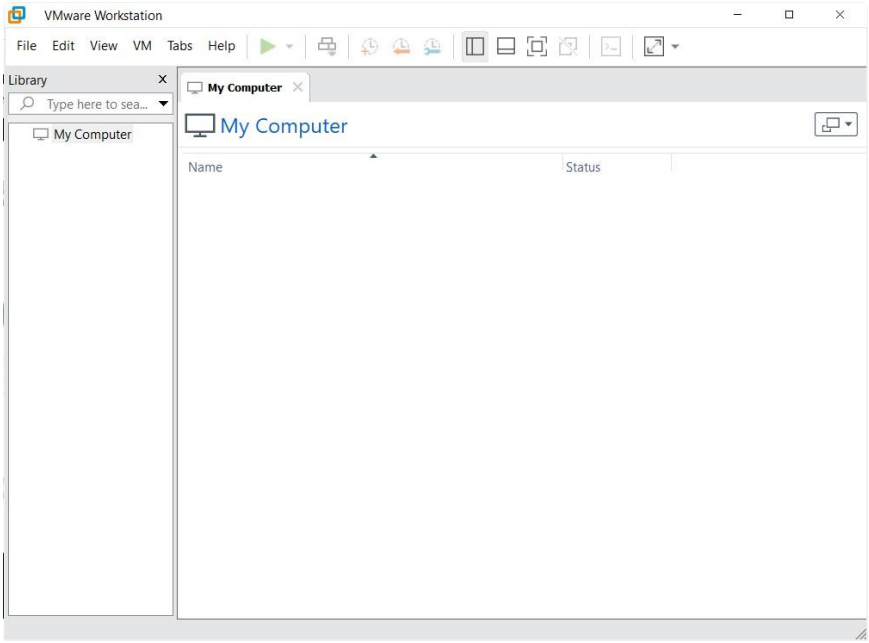
- **Installation and configuring the wazuh in VMware:**

Select your own Virtual Machine: VMware workstation 16 pro.

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

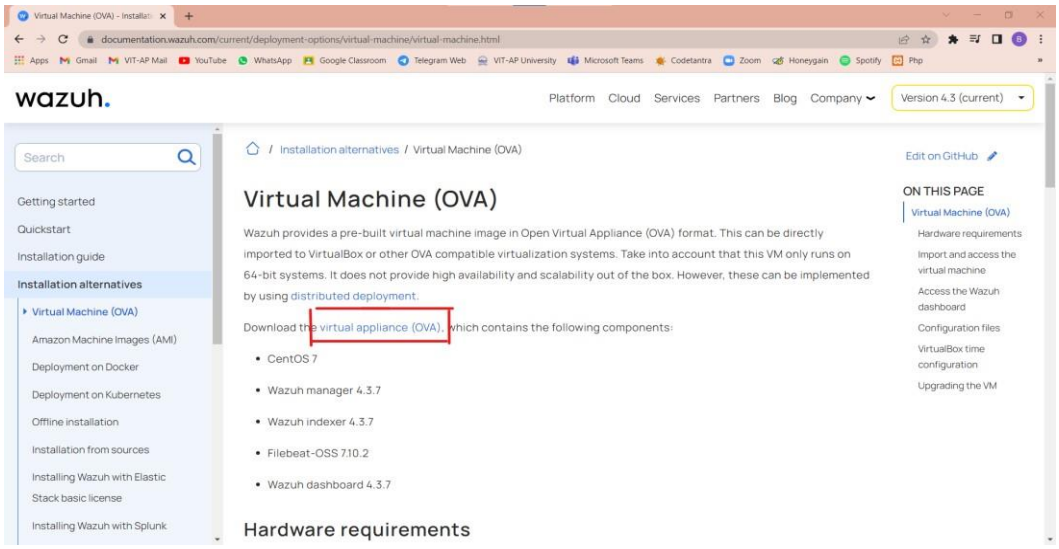
After installation of VMware workstation 16 pro Installation.

- Open the **VMware workstation 16 pro Installation.**

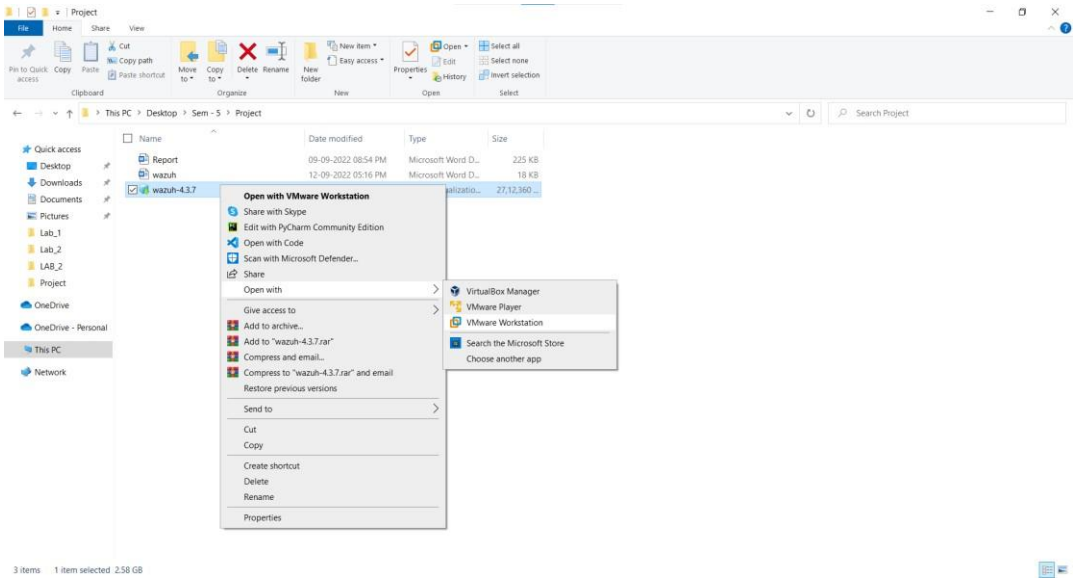


Download the **wazuh-4.3.7.ova**

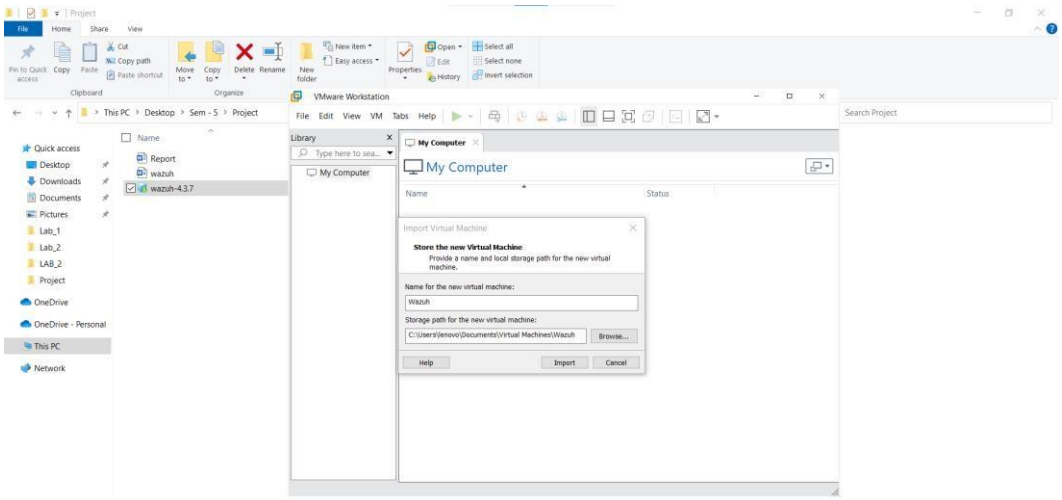
Link: https://documentation.wazuh.com/current/deployment-options/virtualmachine/virtual-machine.html



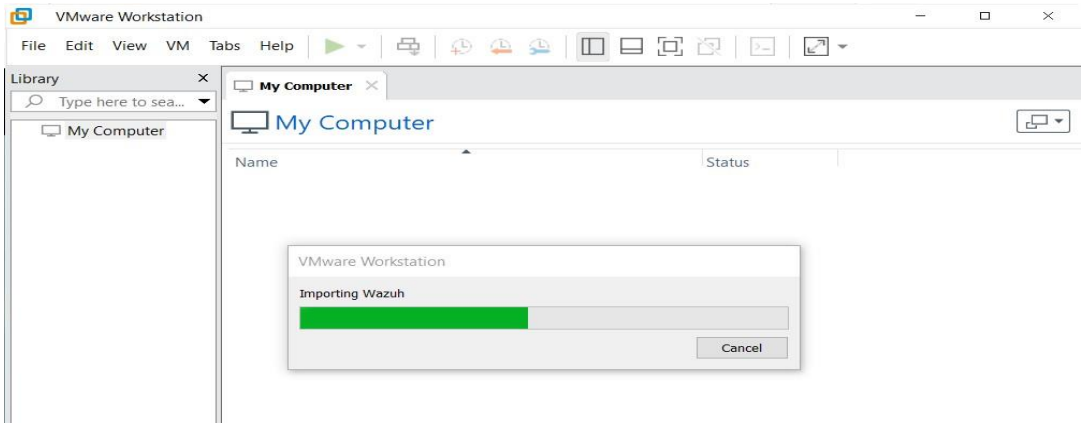
- Open the Wazuh-4.3.7.ova file with VMware workstation 16 pro.



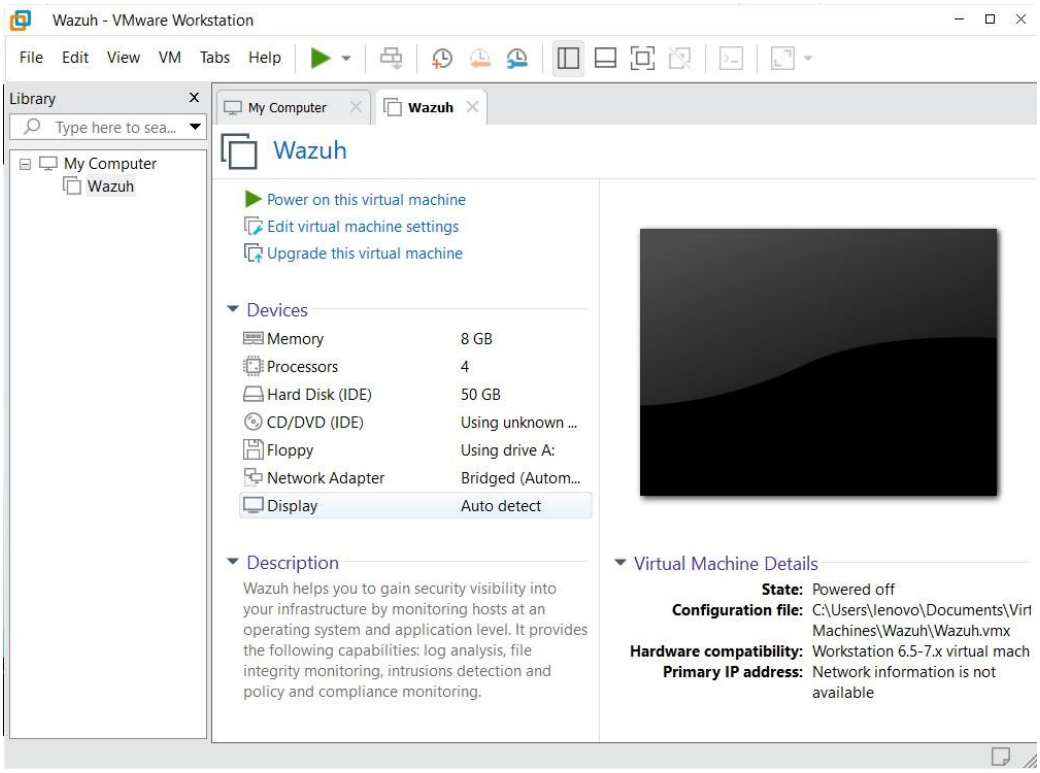
Step 1: Enter the Virtual machine name and leave Storage path for default: Select Import.



- It takes a while!!!!



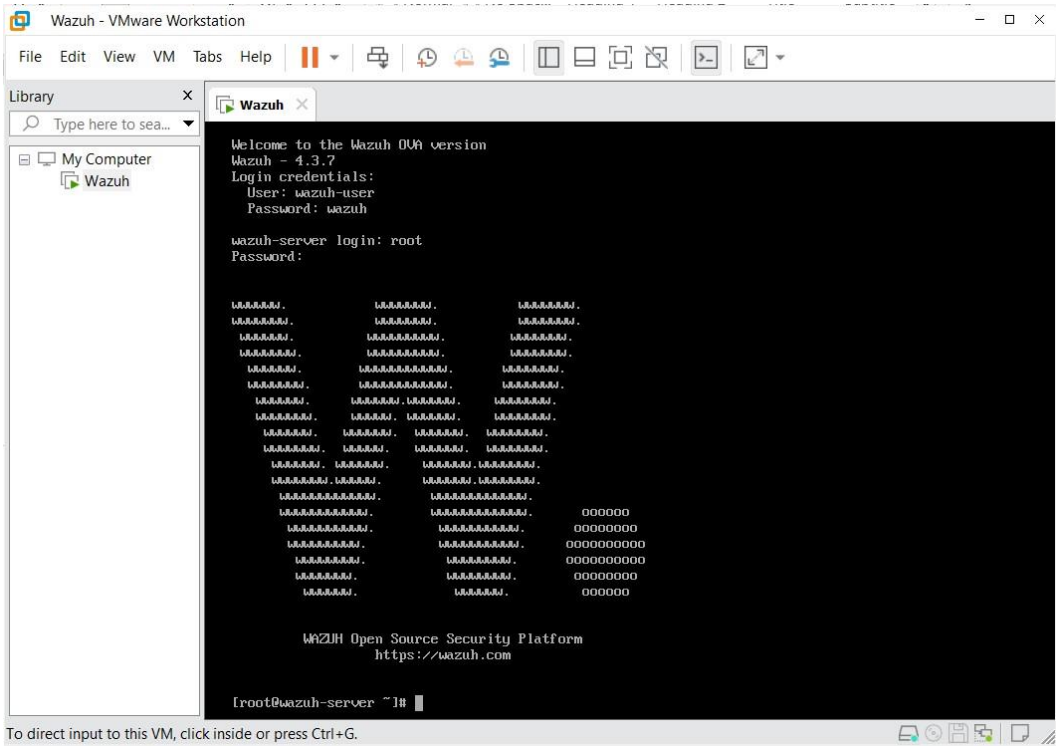
Step 2: After adding the Wazuh to VMware: **Power on this virtual machine**



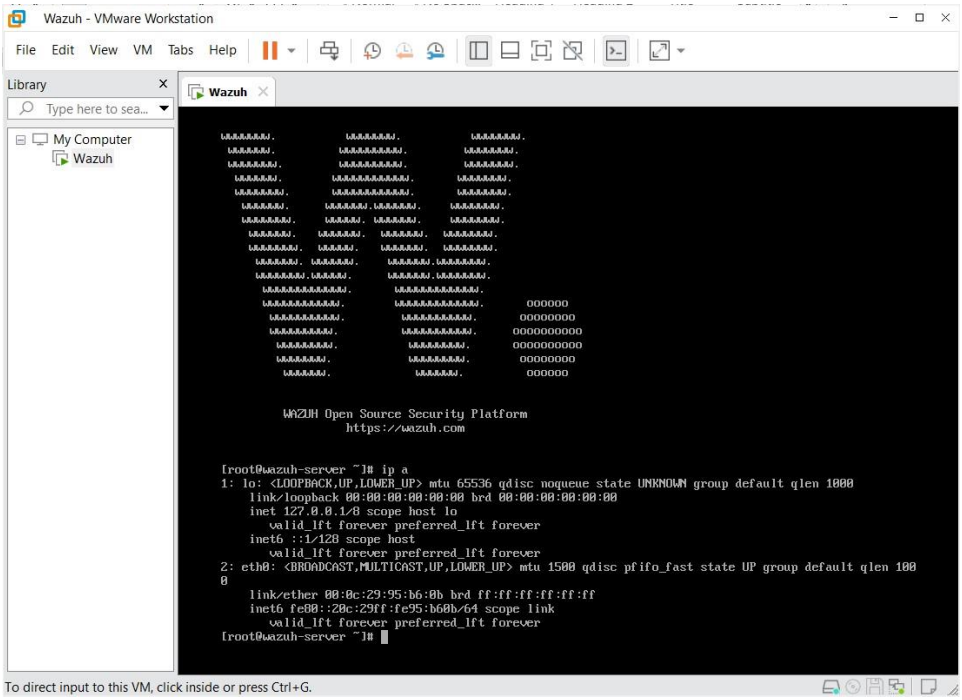
Step 3: Enter the wazuh-server login details:

Username: root

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



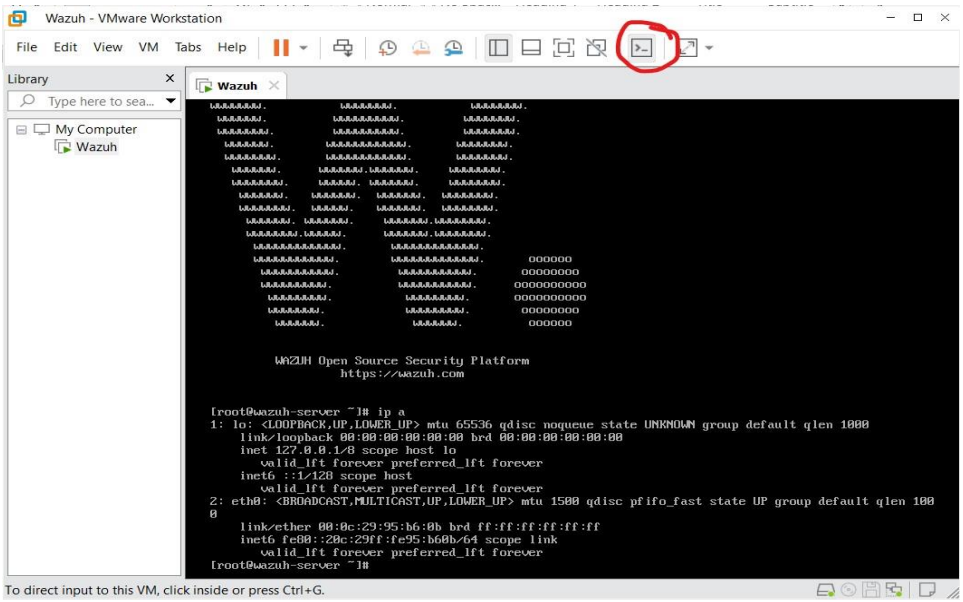
Step 5: Run the command “Ip a” to check the weather server is producing the Ip. If the server is not producing any Ip change the Network setting in the Networks)



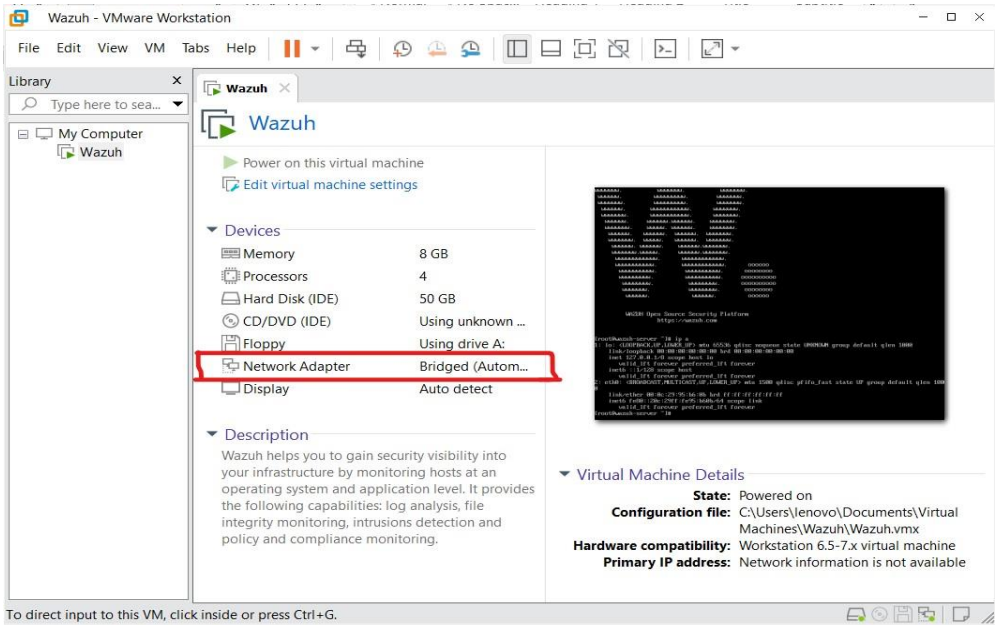
Step 6: Please ignore above steps if you got Ip(Only roman number): Continue there after!

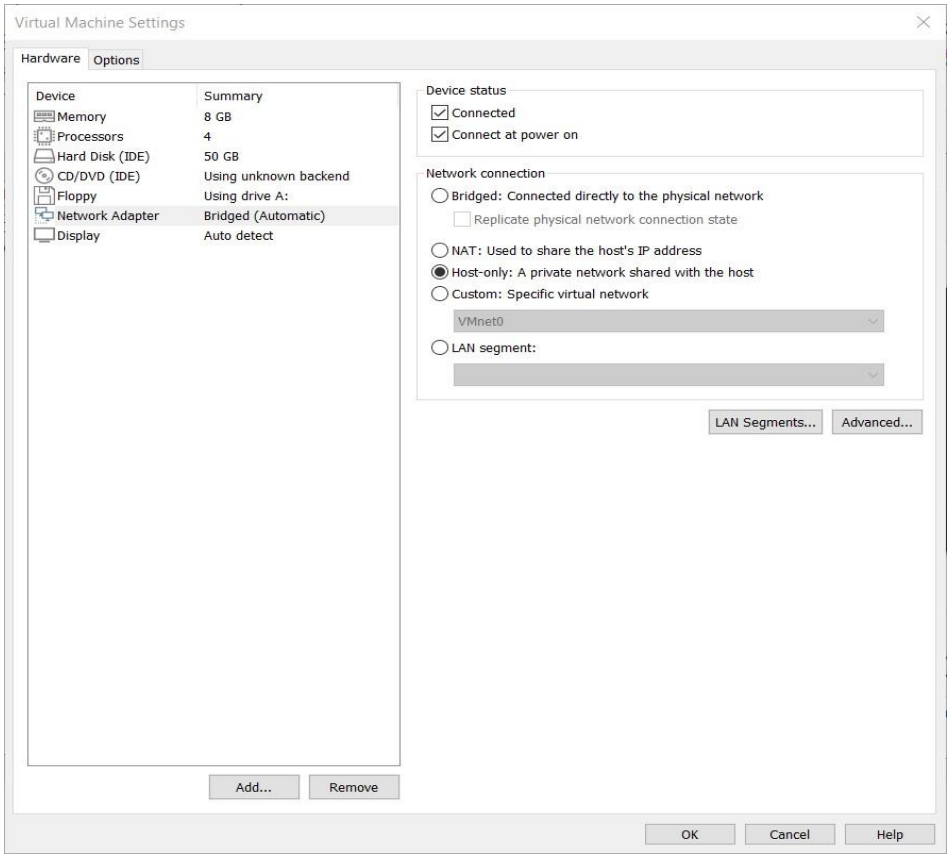
In above server there is no Ip available because the server was running in vm which it was having different network. So, we need to change the network to the host-only network.

- Select the **Show or hide the console view**



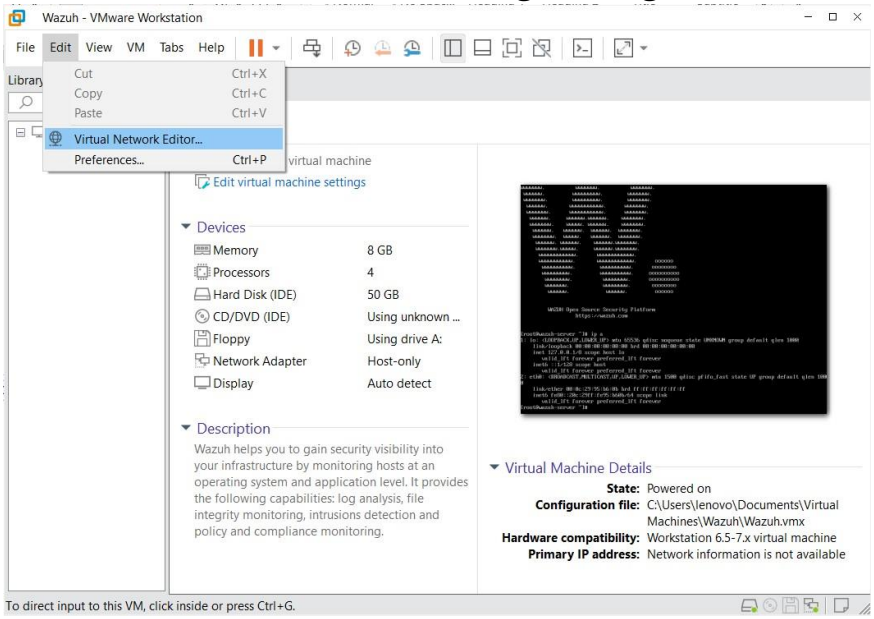
➢ Select Network Adapter to change the Network: **Select the Host-only Network**

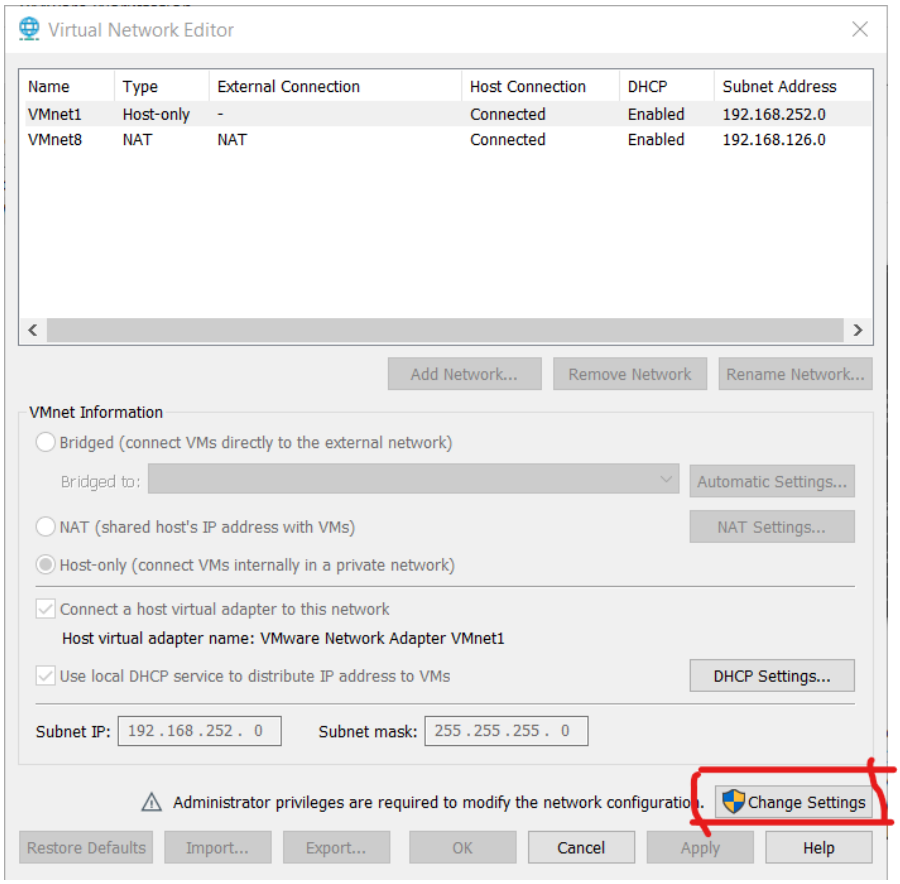




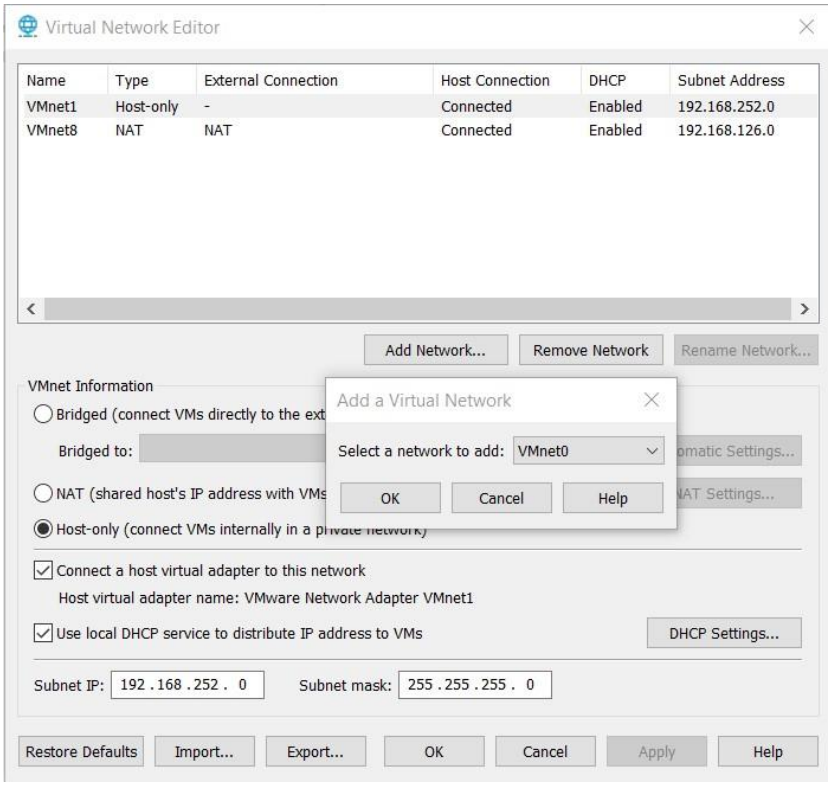
- Turn off the wazuh-server and edit the Network settings

  Edit<Virtual Network Editor<change settings

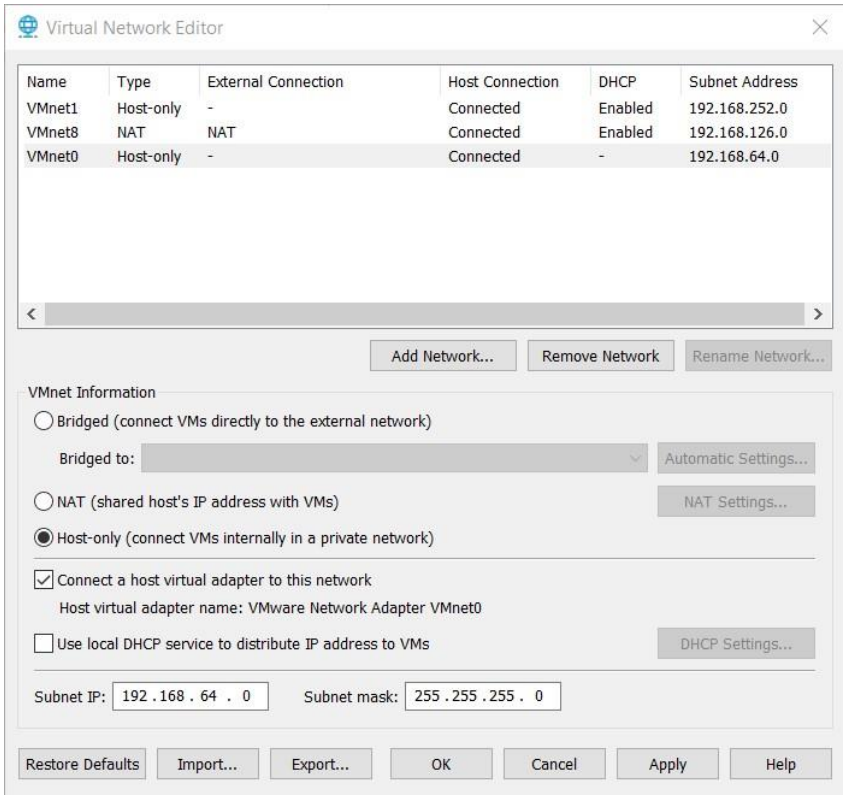




- Add network<VMnet0<ok!



- 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 100
0
    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 ~]#
```

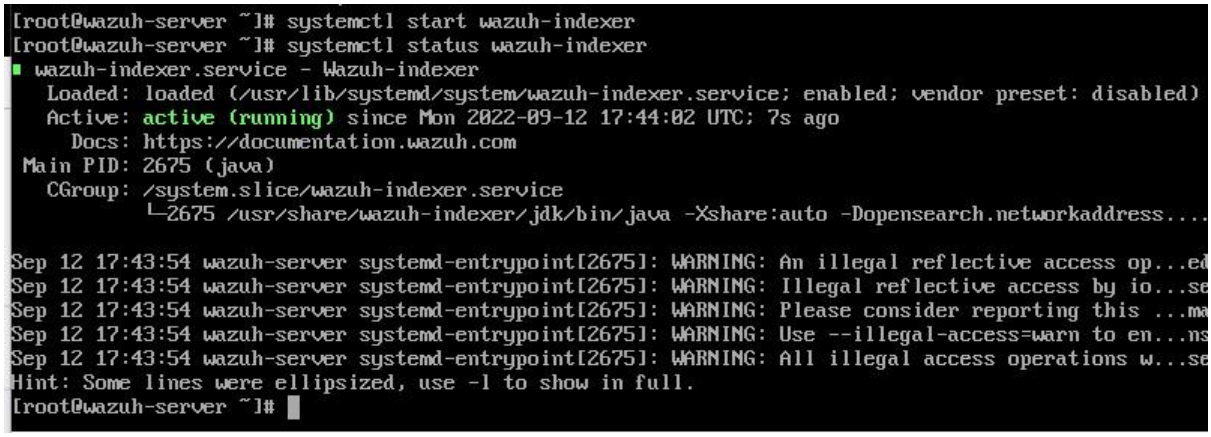
- Run the command to start the wazuh-manager " # **systemctl start wazuh-manager** ")

```
[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/SUCC
ESS)
   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-monitord
           ├─1548 /var/ossec/bin/wazuh-modulesd
           └─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-monitord: Process 1648 not used by Wazuh, removing...
Sep 12 23:10:54 wazuh-server env[752]: Started wazuh-monitord...
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")

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

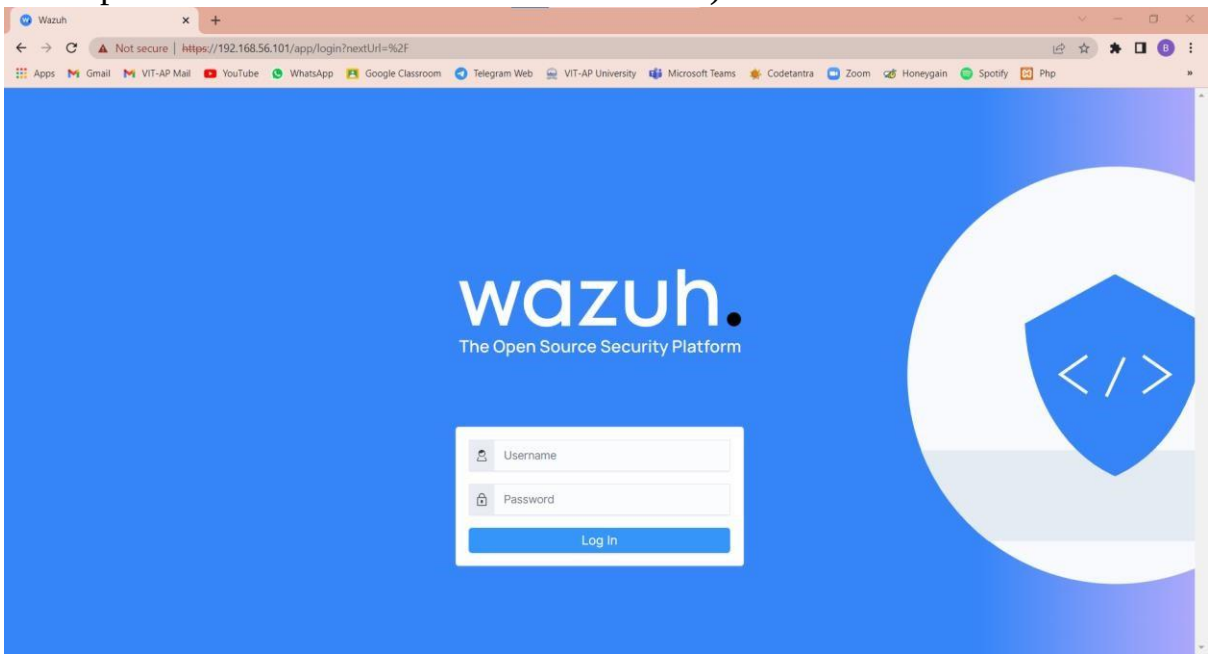
  Command: 1.Systemctl enable wauh-indexer
  2.Systemctl start wazuh-indexer
  3.Systemctl status wazuh-indexer



❖ Some useful commands for wazuh server:

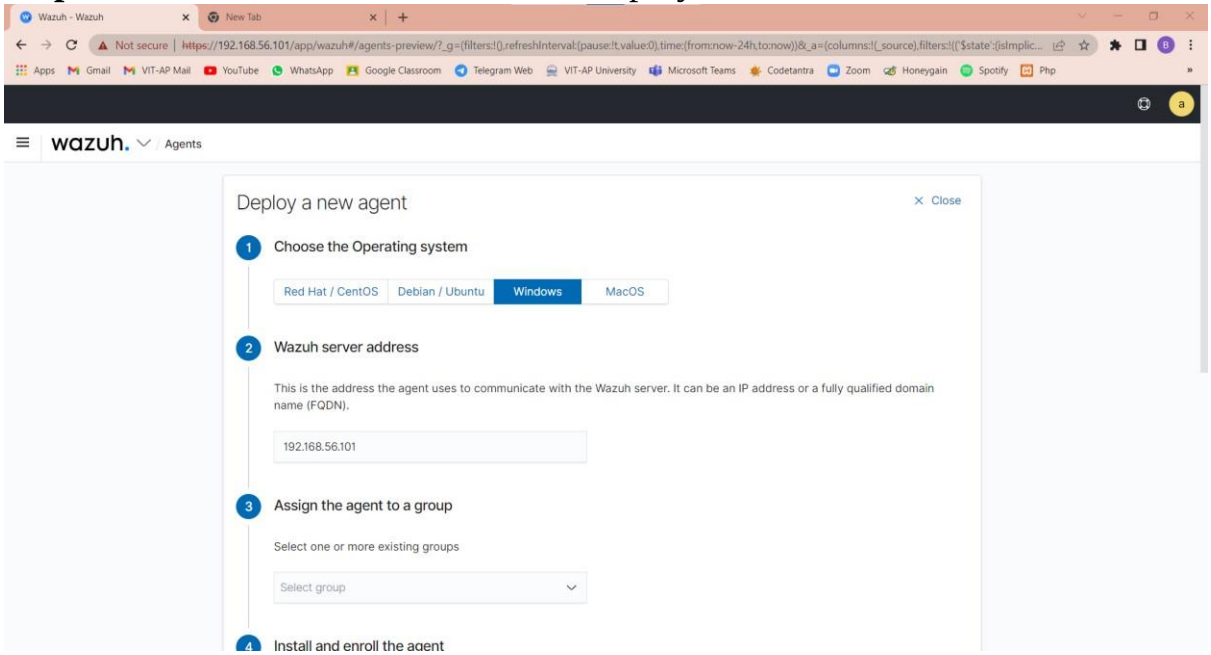
start	Start the Wazuh processes.	
stop	Stop the Wazuh processes.	
restart	Restart the Wazuh processes.	
reload	Restart all Wazuh processes except wazuh-execd. This allows an agent to reload without losing active response status. This option is not available on a local Wazuh installation.	
status	Determine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

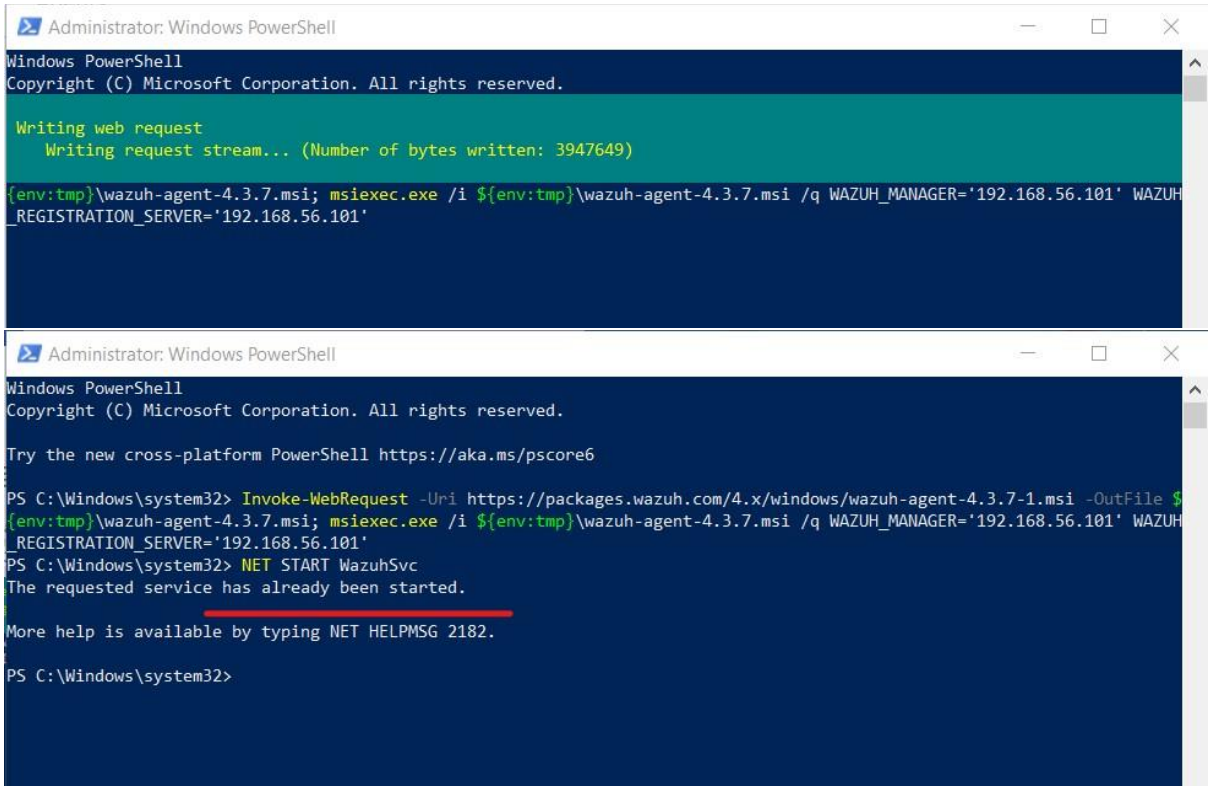


Deploying the agents to monitor the security activity.

Step 1: Select add agents
Step 2: select the OS which need to be deployed.

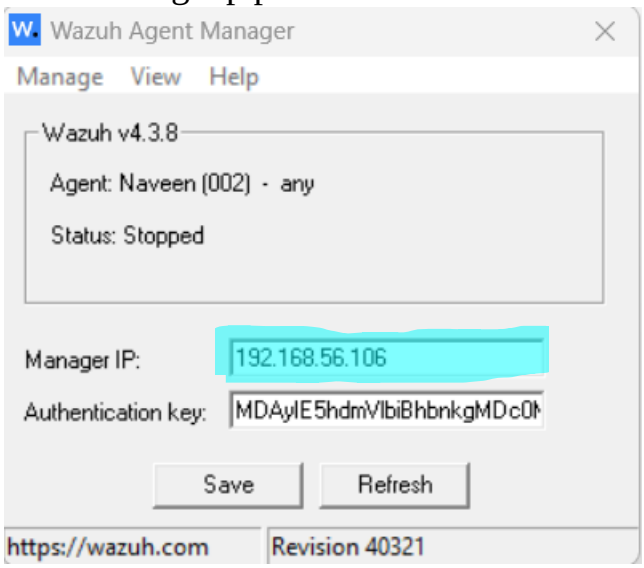


Step 2: Open the Windows Power Shell with administrator and the link that are in step 4 and 5.

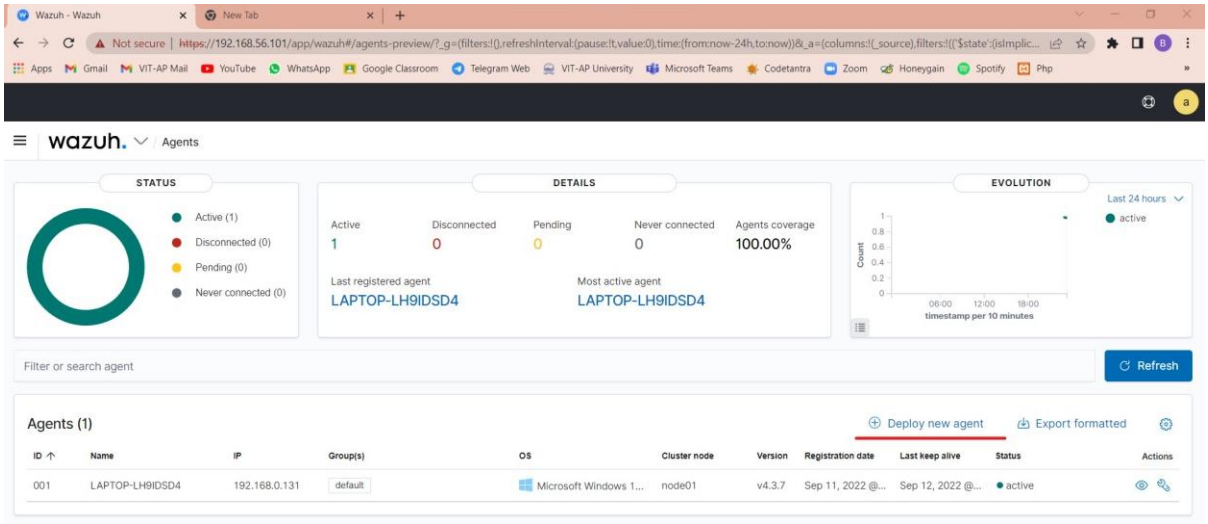


Alternative way to add the agents.

- 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.

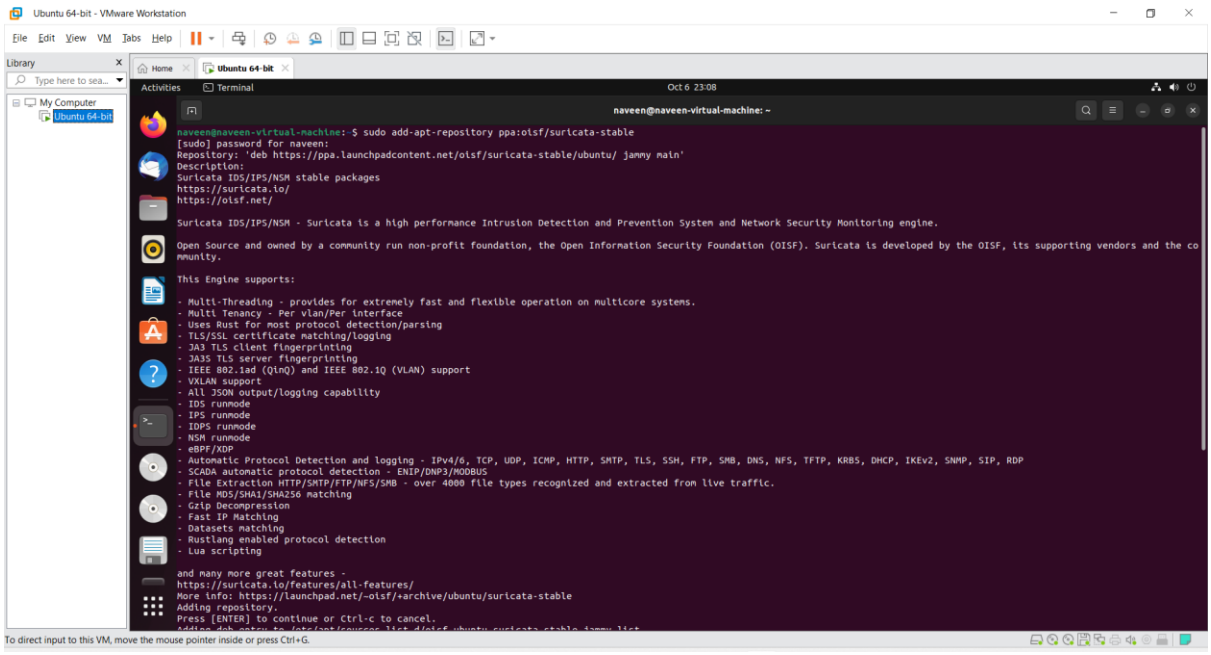
Use of Suricata?

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

Installation:

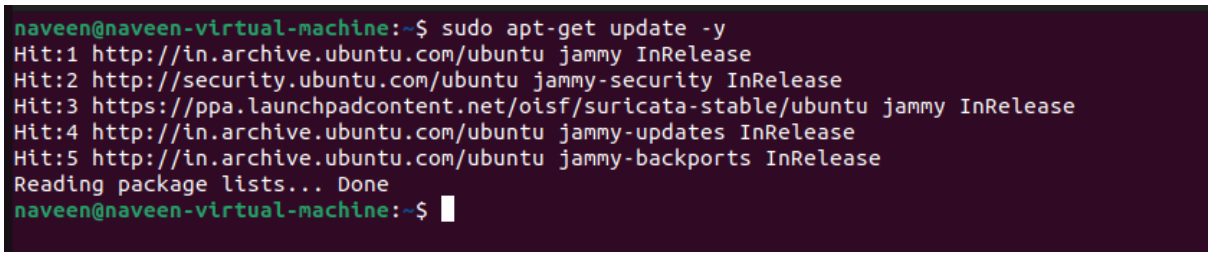
Step 1: Open the ubuntu terminal

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

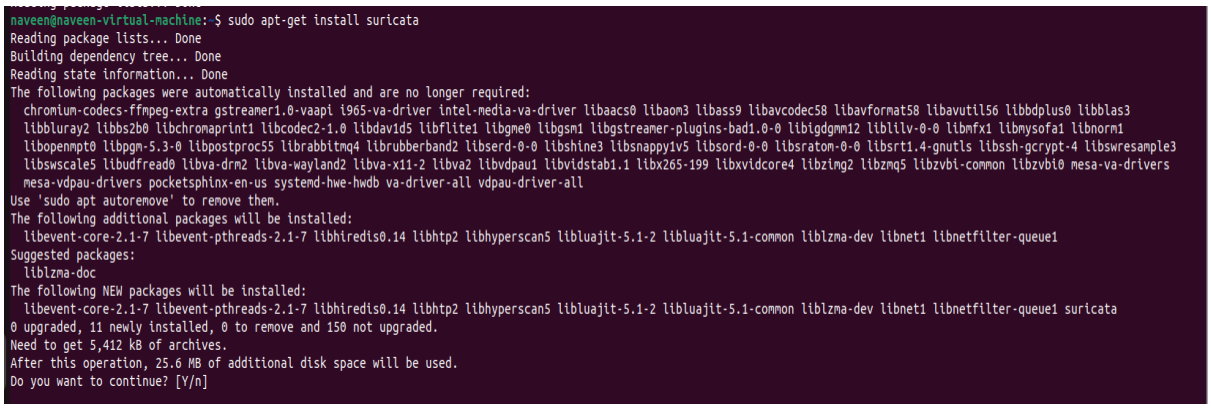


Step 3: Update the terminal by running the command " sudo apt-get update -y "

.

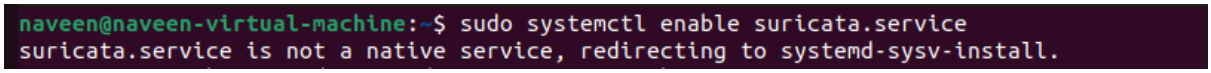


Step 4: Install the suricata by using the following command "sudo apt-get install suricata"



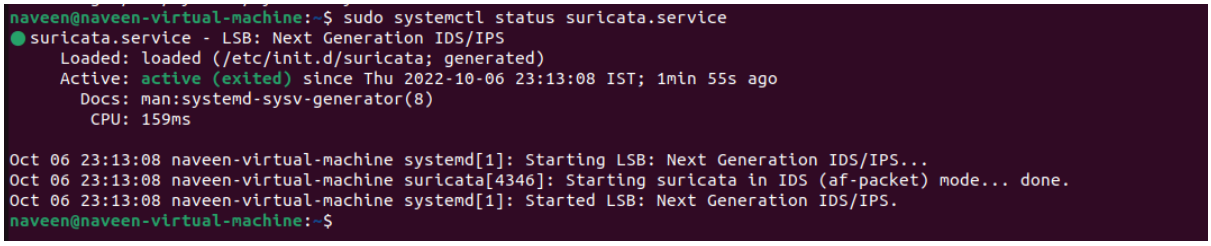
Step 5: After installing the Suricata successfully .Let's start the suricata service by running the following commands.

-" sudo systemctl enable suricate.service "

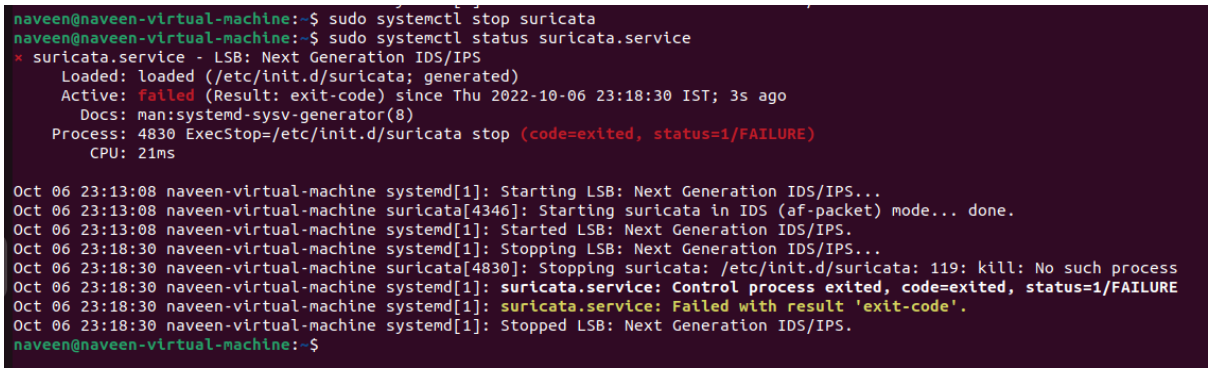


-" sudo systemctl start suricata.service "

-" sudo systemctl status suricata.service "



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 "



- The files and configuration file be stored in suricate " 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:~$
```

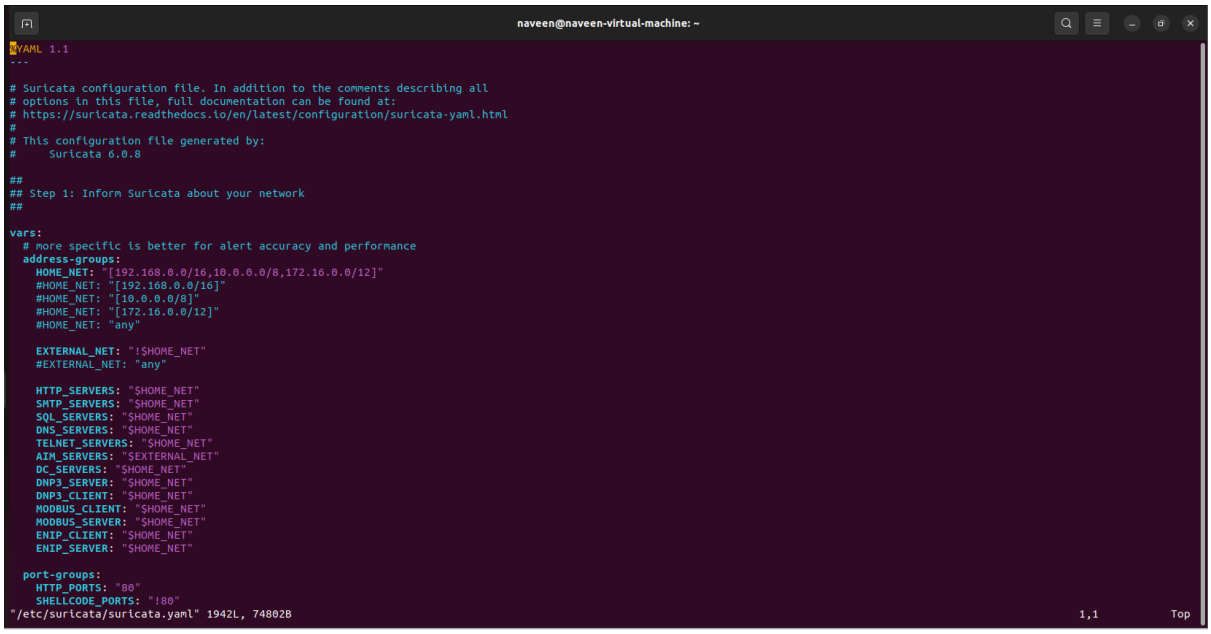
Step 7: Install the vim in the Ubuntu by running the following command

" 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 i965-va-driver intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas3
  libbluray2 libbs2b0 libchromaprint1 libcodec2-1.0 libdav1d5 libflite1 libgme0 libgsm1 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libmfx1 libmysofa1 libnorm1
  libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0 libsrt1.4-gnutls libssh-gcrypt-4 libswresample3
  libswscale5 libudfread0 libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 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:
  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.

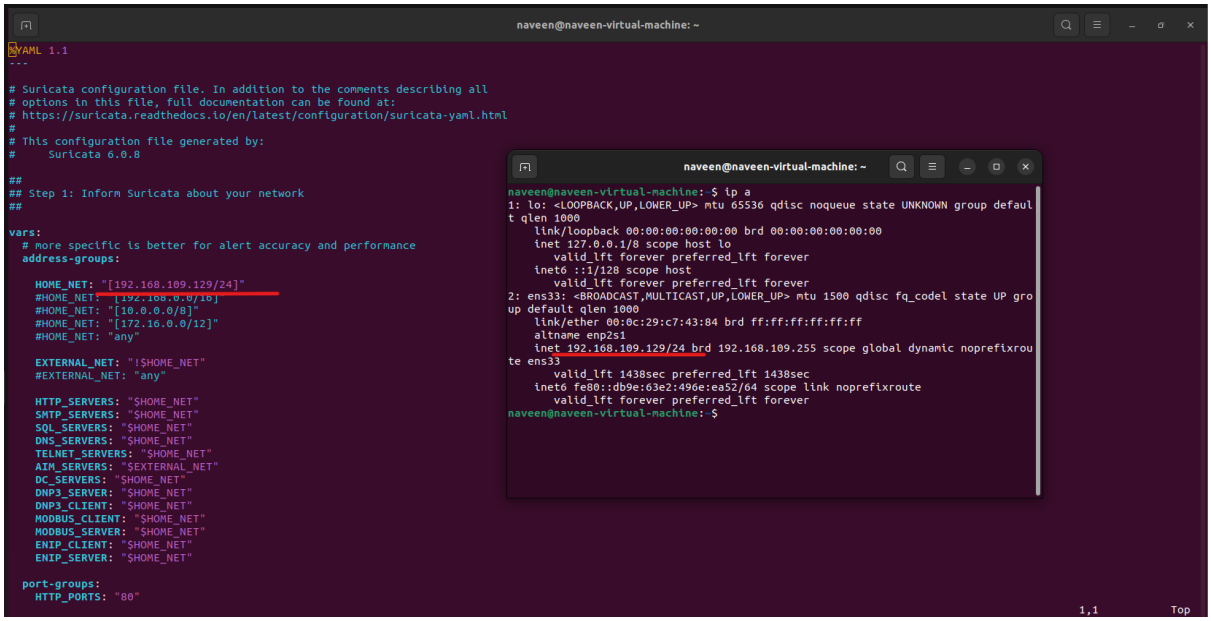
" sudo vim /etc/suricata/suricata.yaml "



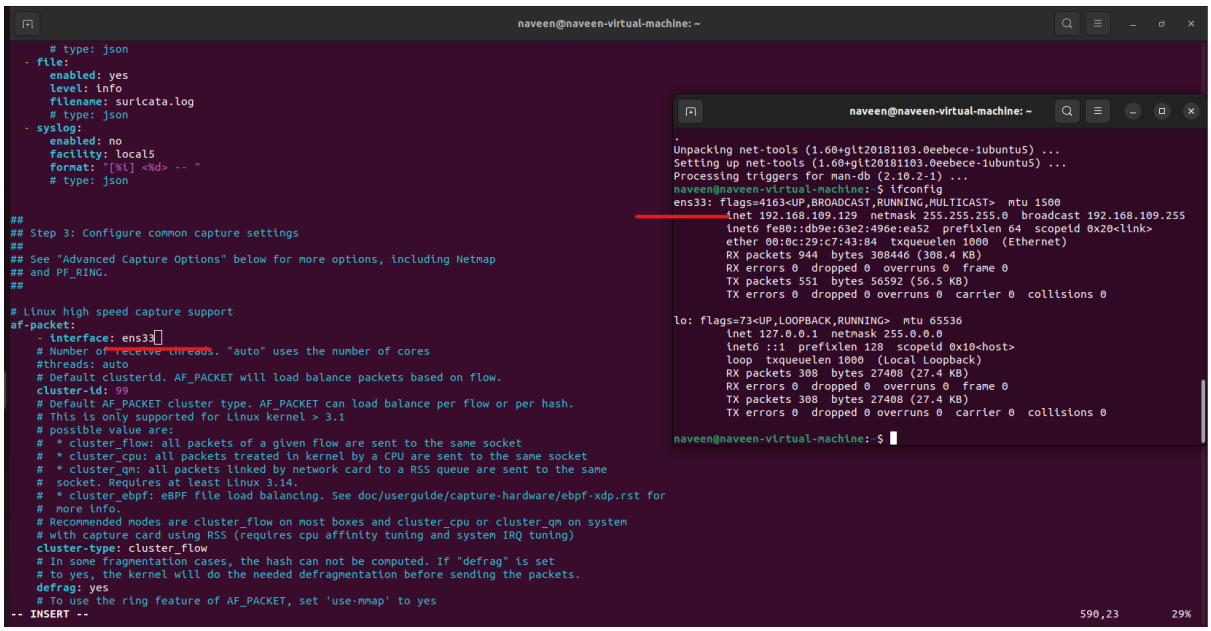
- For searching a word in yaml file use "/"
- For inserting the data in yaml file use "ctrl+o"
- For saving the data in yaml file use "esc, ctrl+: , wq"
- For quitting from a yaml file use ":q"

Step 9: Change the HOME_NET details to you laptop's ip with subnet range

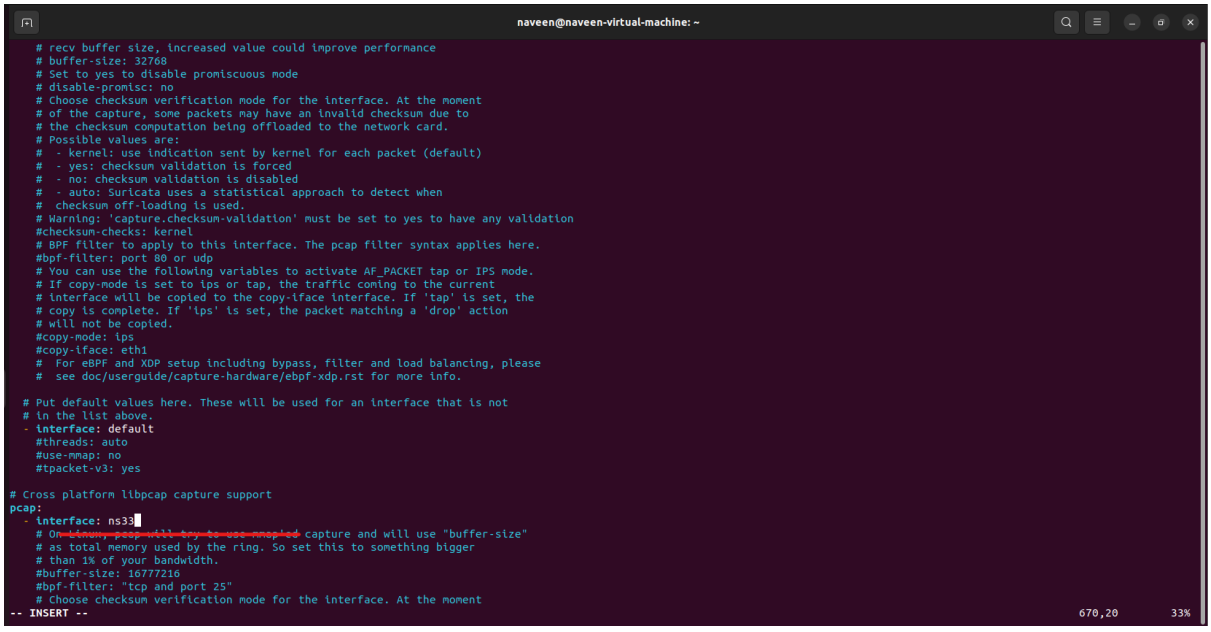
(you can get the HOME_NET ip by using the above command "ip a")



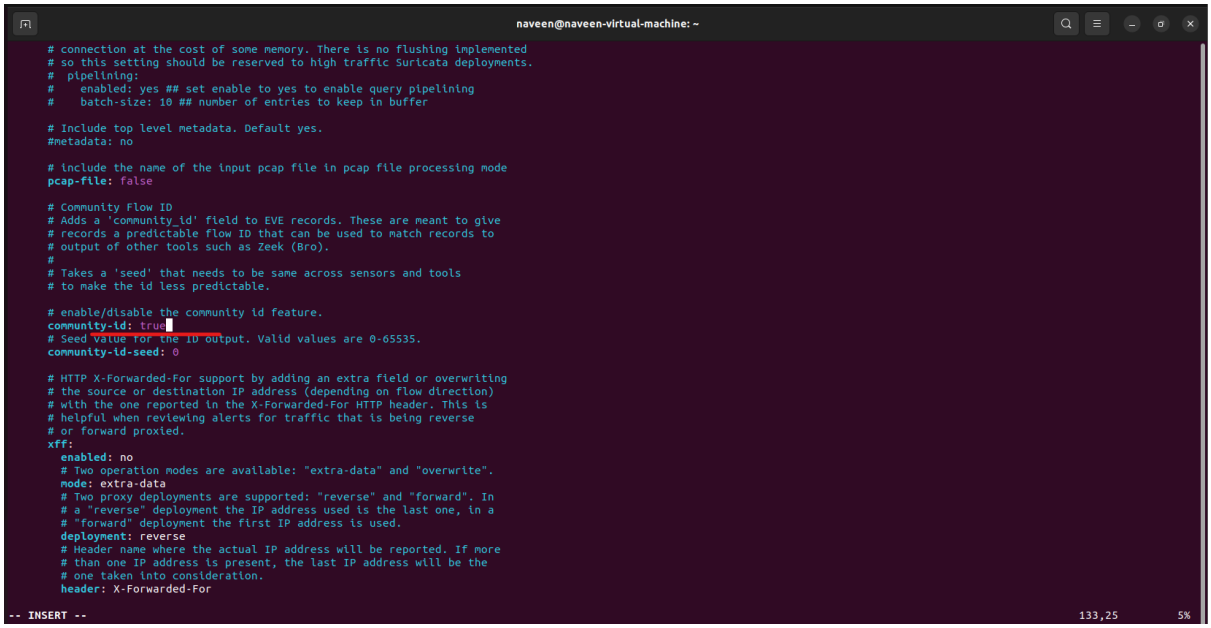
Step 10: Search for “af-packet” in yaml and change the interface name according to your internet interface:



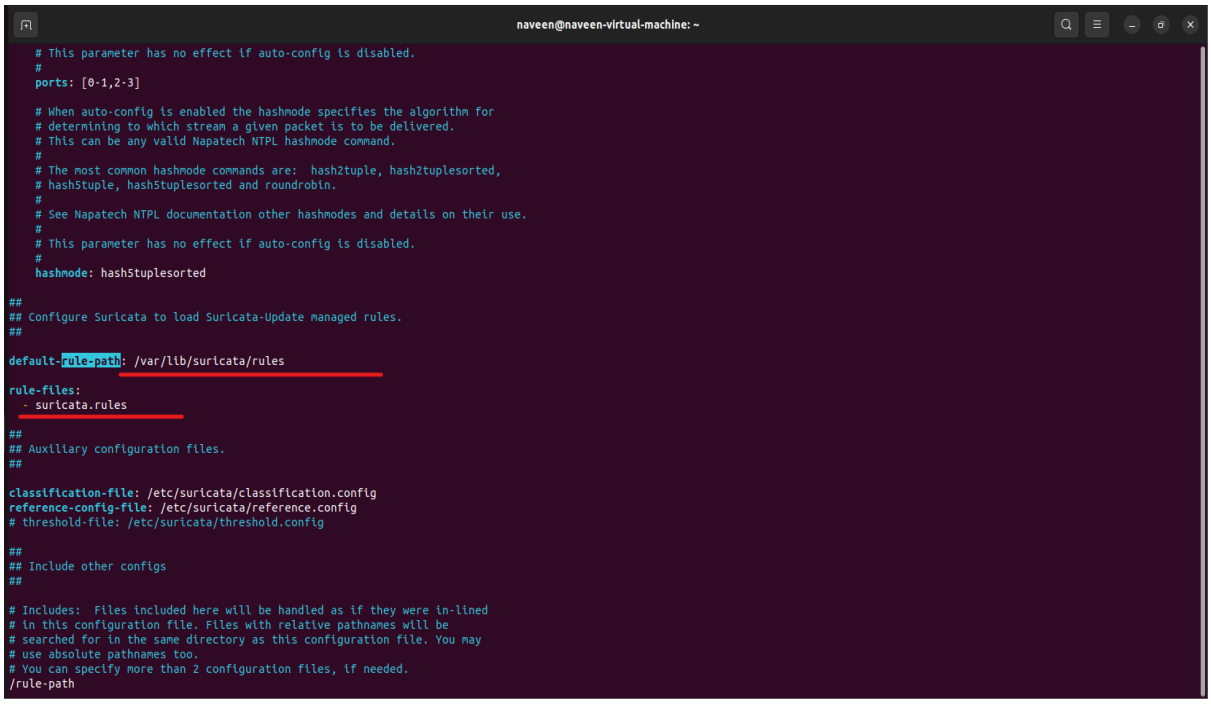
Step 11: Search for “pcap” in yaml file and make a change in interface



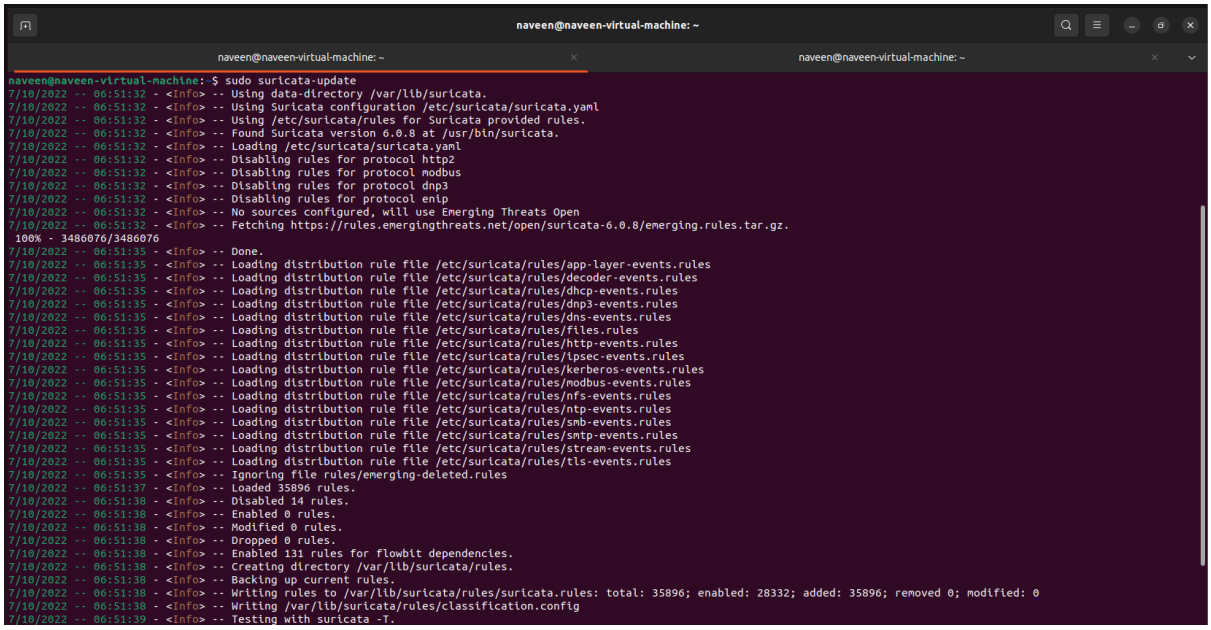
Step 12: Under pcap-file set the community-id as true:



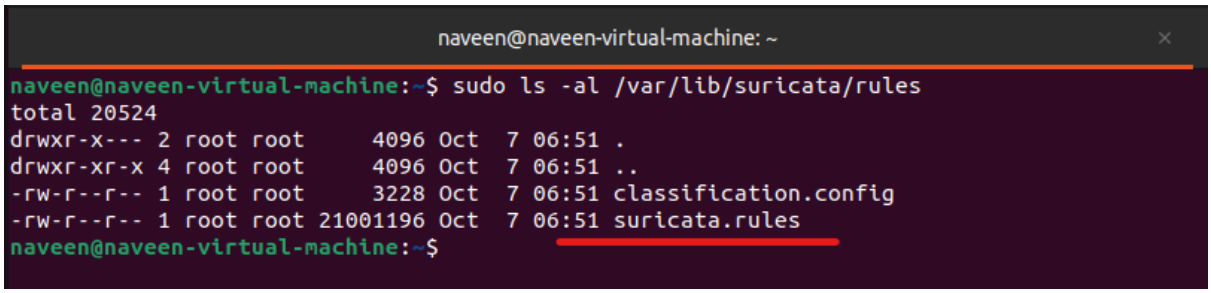
Step 13: Search for the rule-path and make sure that above details are present:



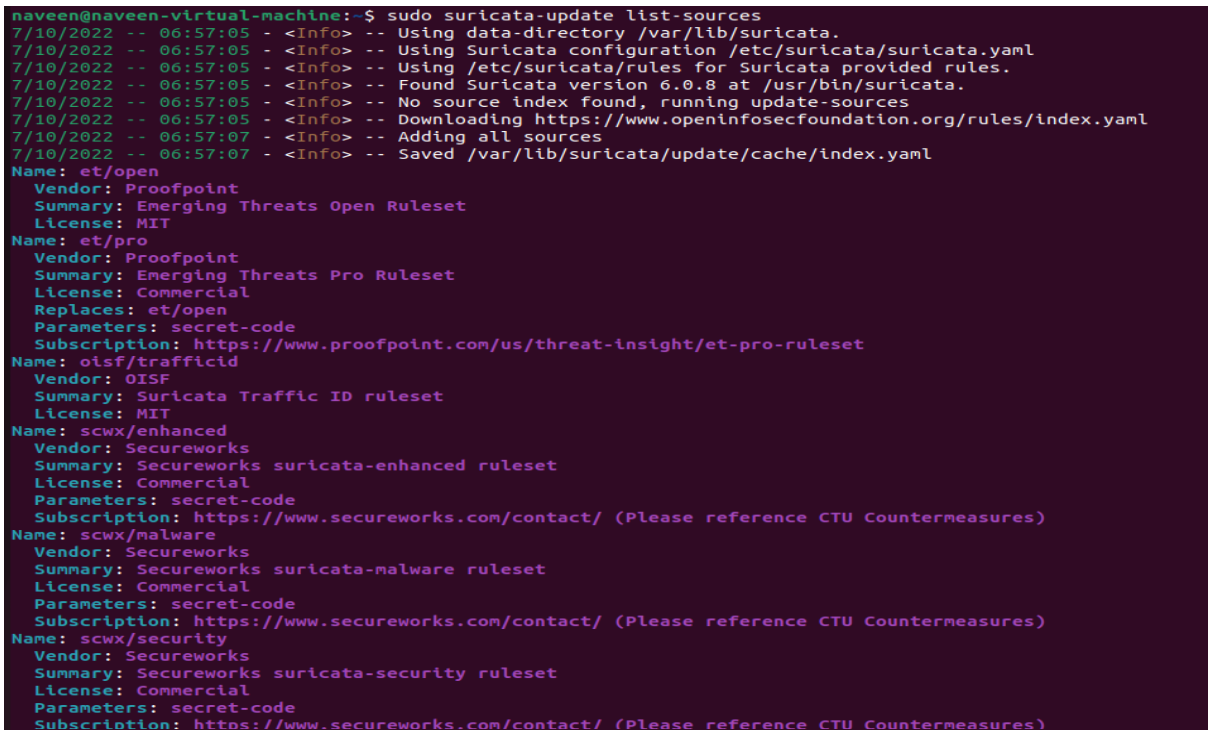
Step 14: Update the suricata file for seeing the by default rule files: (skip the warning if you got it)



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



- If you want to update the suricata rules with other rules that are present in open sourse use the above command: "sudo suricata-update list-sources"

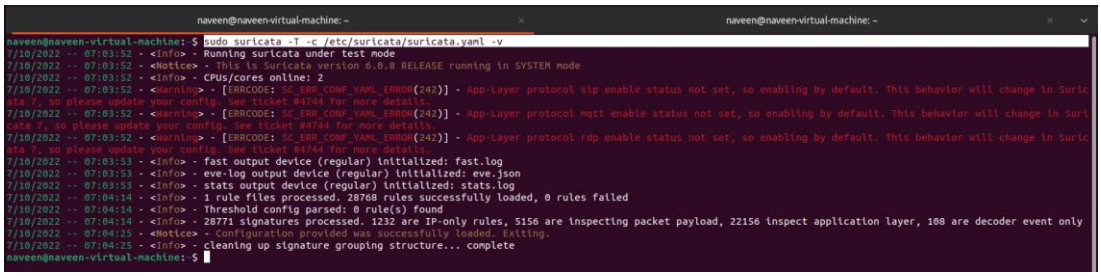


- You can use any of the file that was present in list-sources each sources gives different rules (In my case I was adding the malsilo/win-malware)
- For adding it using the above command: "sudo suricata-update enable-source malsilo/win-malware"

```
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.



Step 15: Start the suricata and check the status:

```
naveen@naveen-virtual-machine: ~
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://testmynids.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.logv "

```
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 [**] [Cl
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

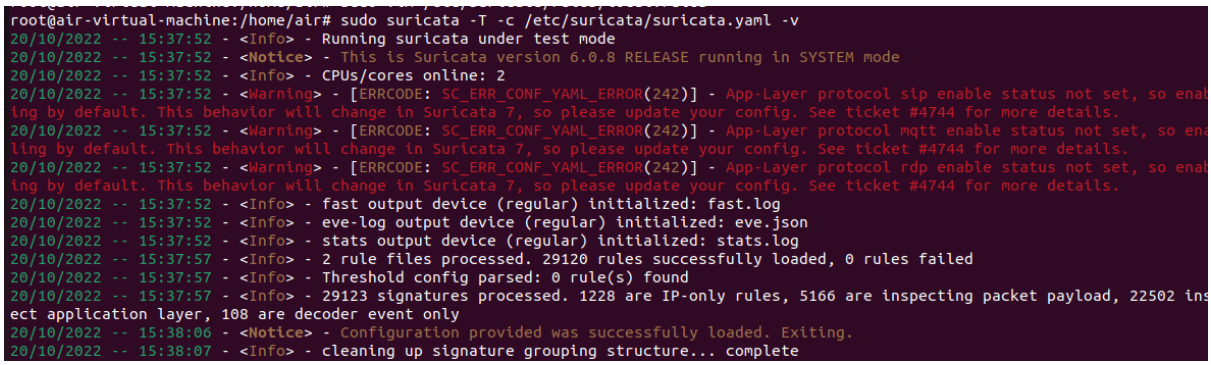
-currently we are adding the icmp rule

Command: " sudo vim /etc/suricata/suricata.yaml "

Add the rule in the rules path

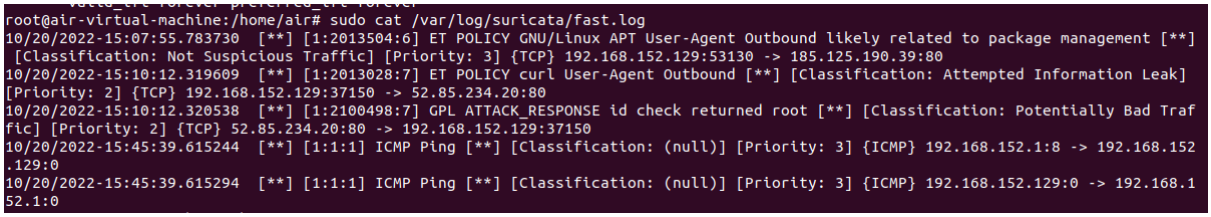
```
default-rule-path: /var/lib/suricata/rules

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/suricta/suricat.yaml –v "



Step 22: ping the Ubuntufrom 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



- 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 "

```
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)
```

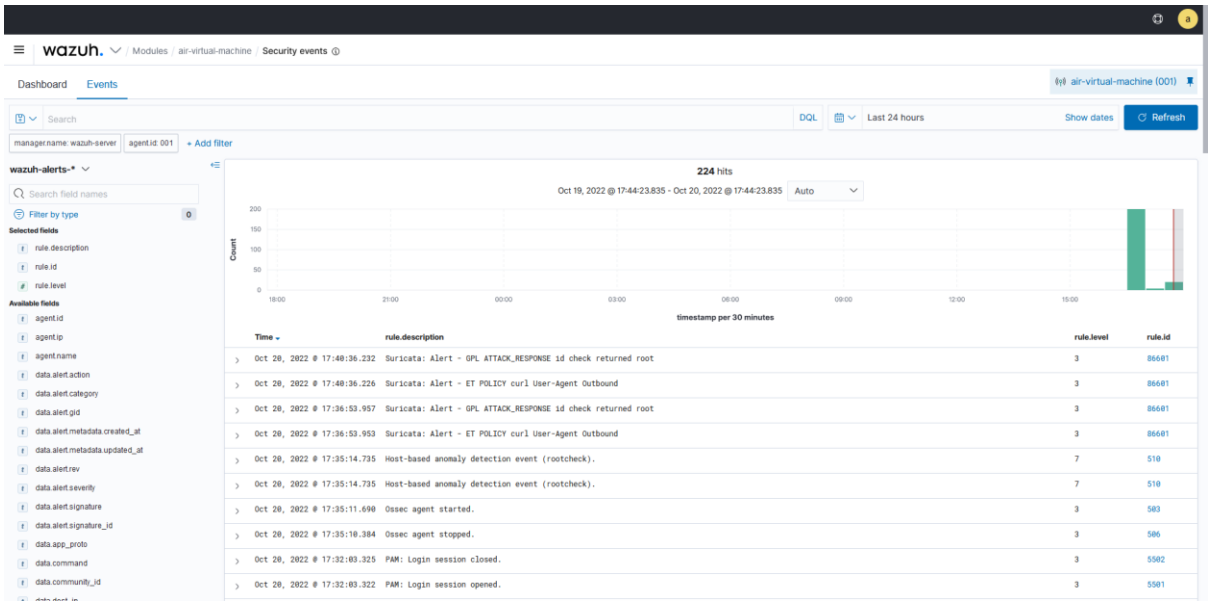
Step 2: Run the following command to capture the logs

" 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_22.04//'sudo tee/etc/apt/sources.list.d/security:zeek.list “

After running the above command ,execute the following command:

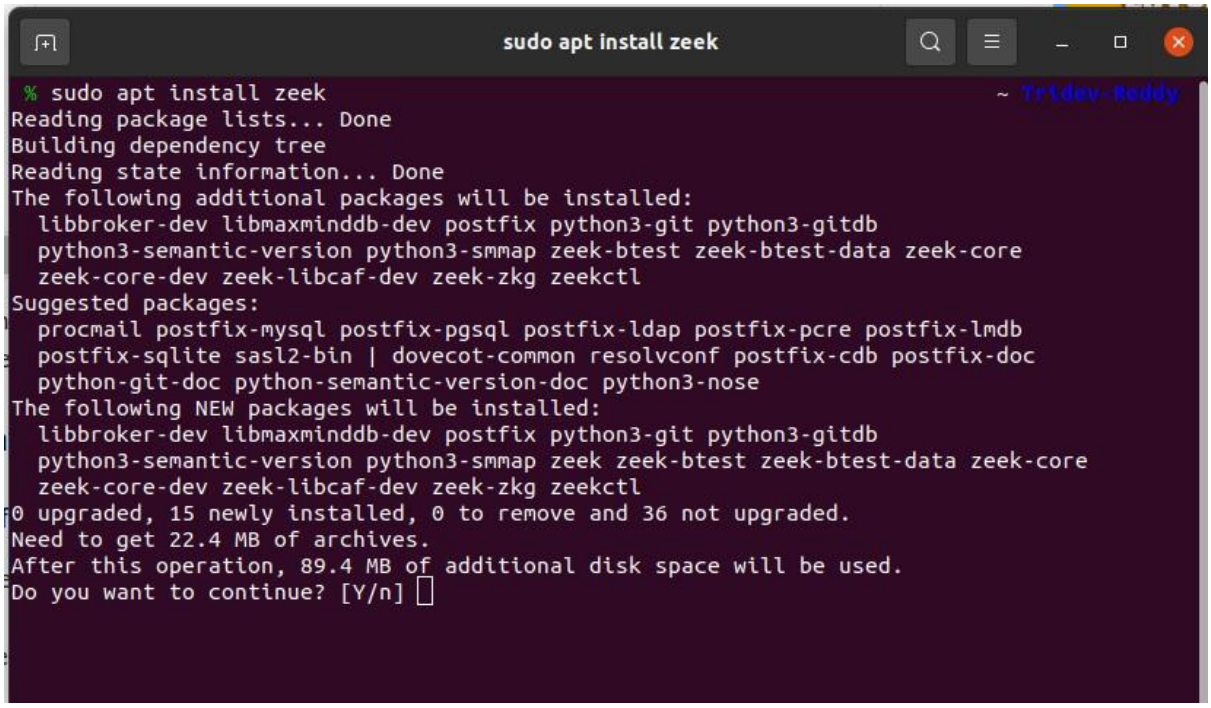
Command: “ curl -fsSL https://download.opensuse.org/repositories/security:zeek/xUbuntu_20.04/Rel ease.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/security_zeek.gpg > /dev/null “

The output looks like:

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



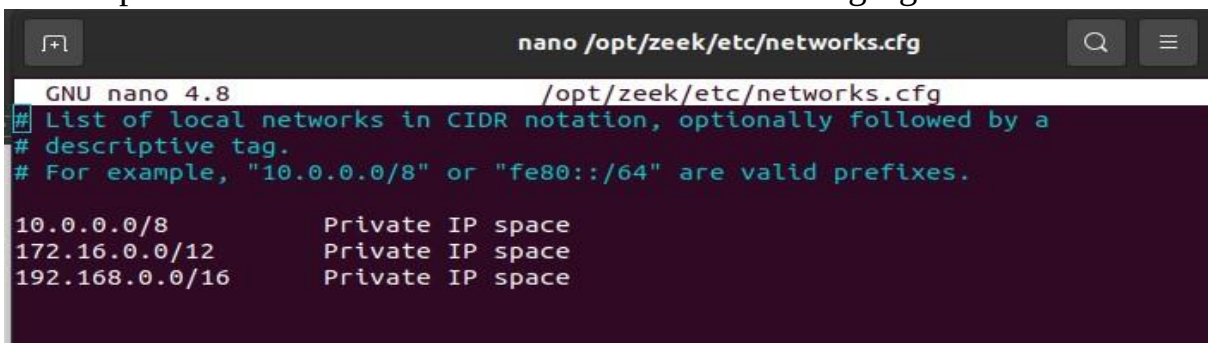
Step 4: Add Zeek binary path to PATH

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.



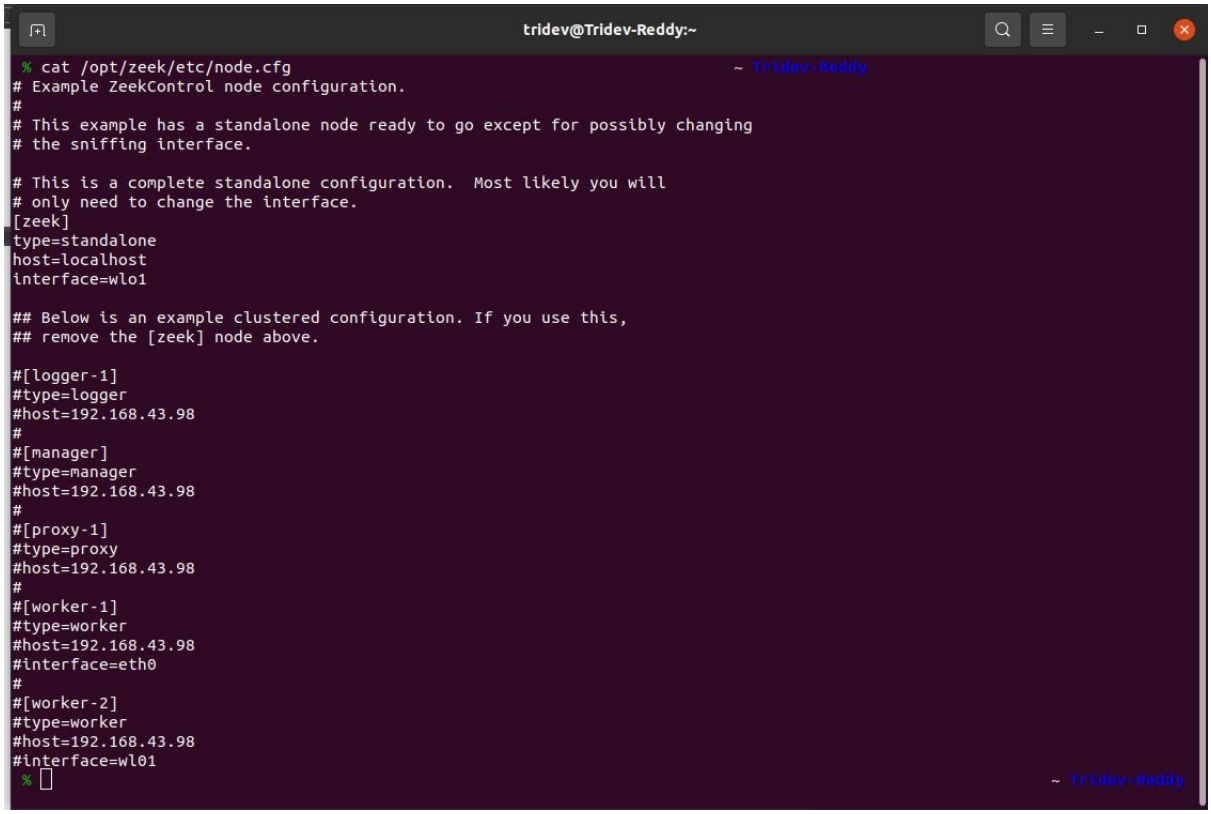
- Zeek can be run in standalone mode or in a cluster setup. To define whether to run in a cluster or standalone setup, you need to edit the /opt/zeek/etc/node.cfg configuration file.

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



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

cd *opt*/zeek/bin

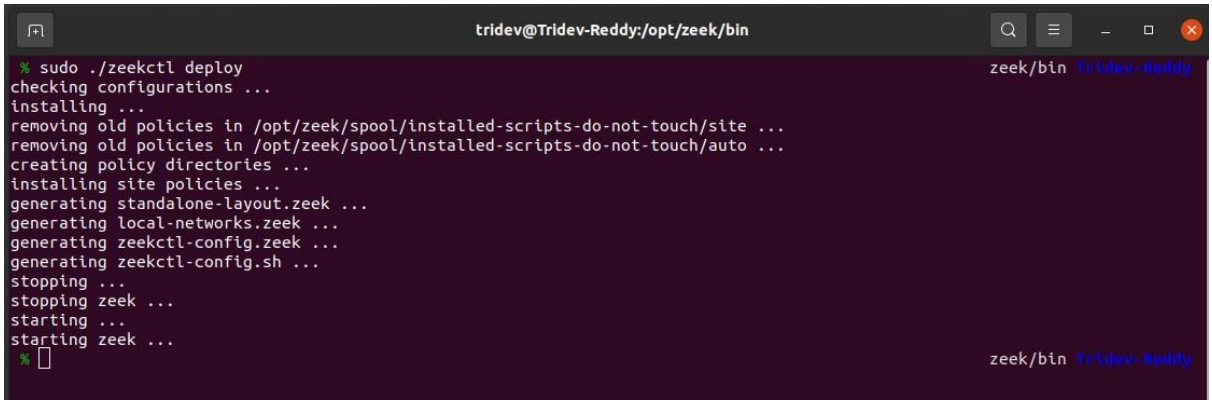
sudo ./zeekctl check

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

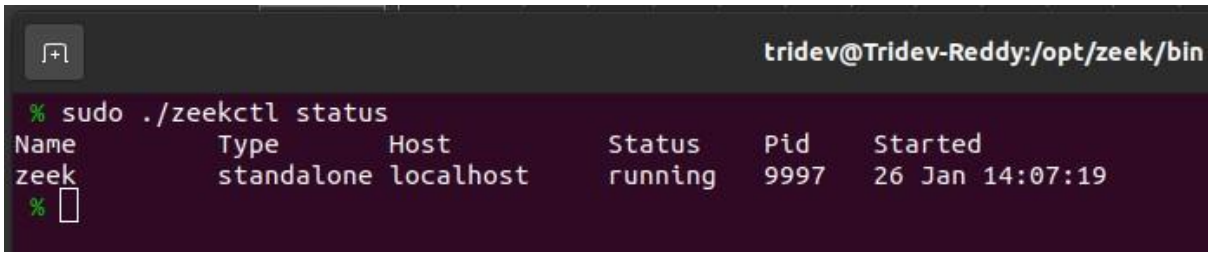
Step 6: Checking zeekctl configuration

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

Command: " sudo ./zeekctl deploy "



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

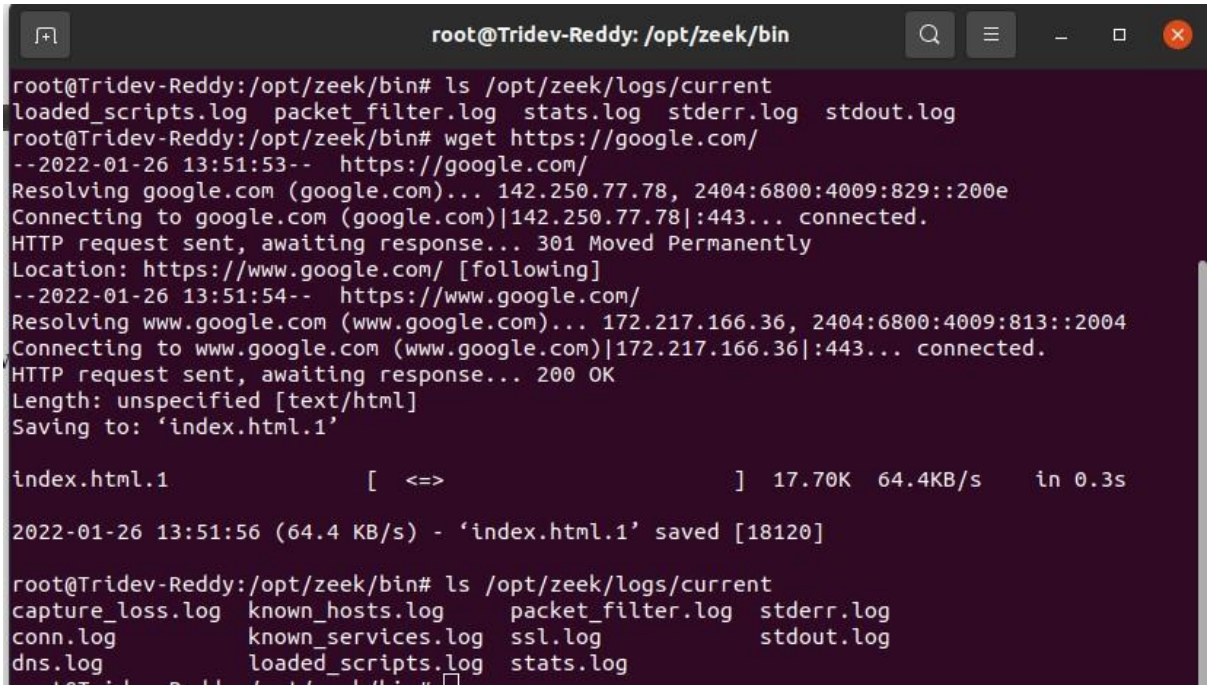


DEMO:

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

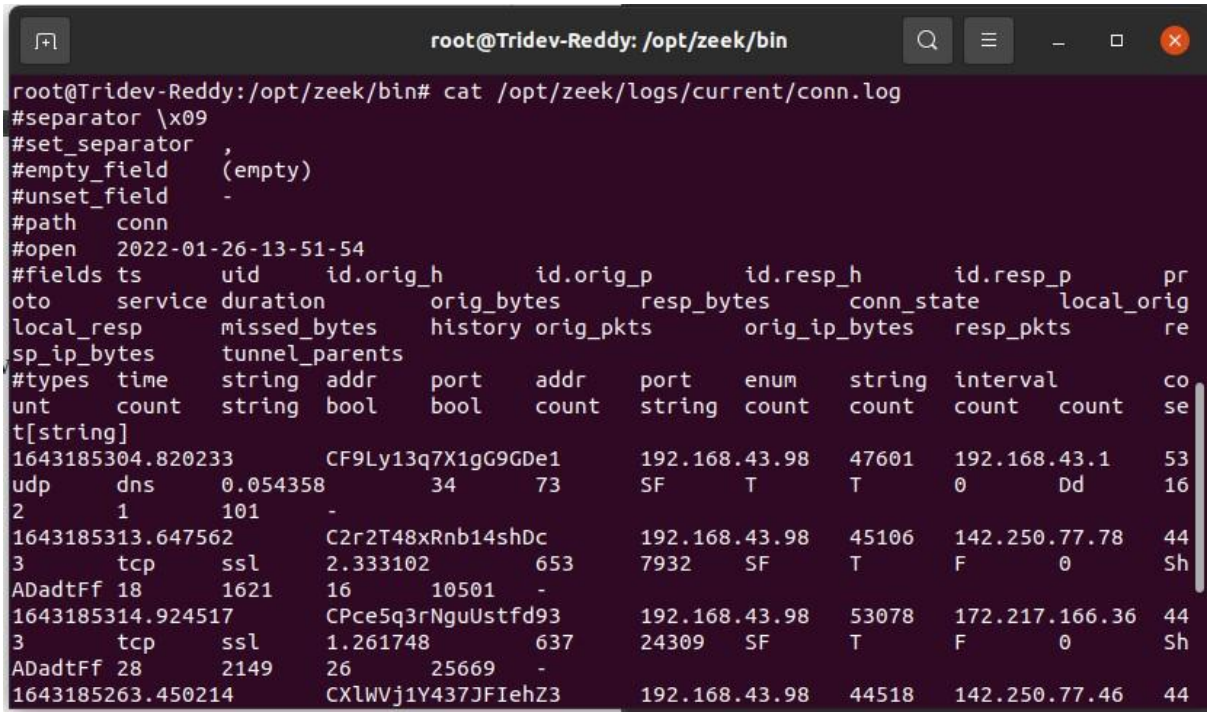
./zeekctl start

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)



- 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.



- 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.

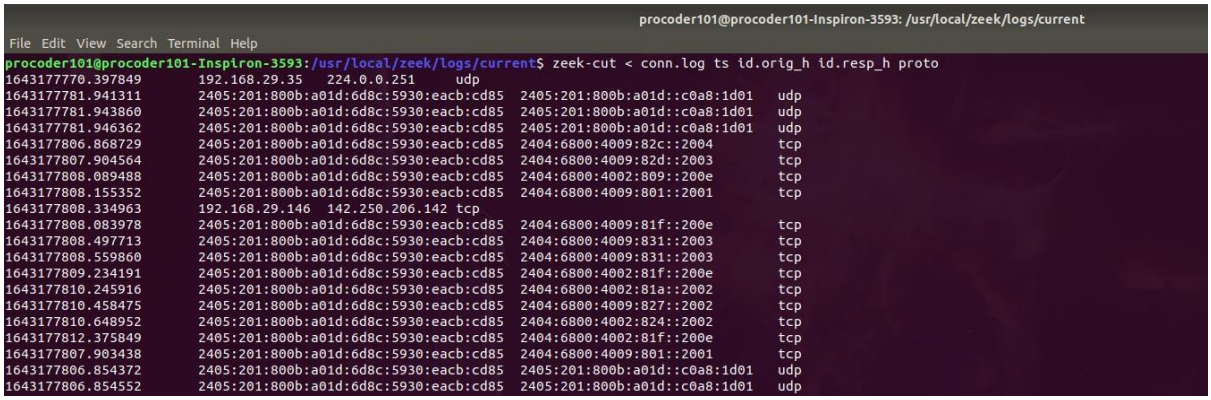
This can be done by running

./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.

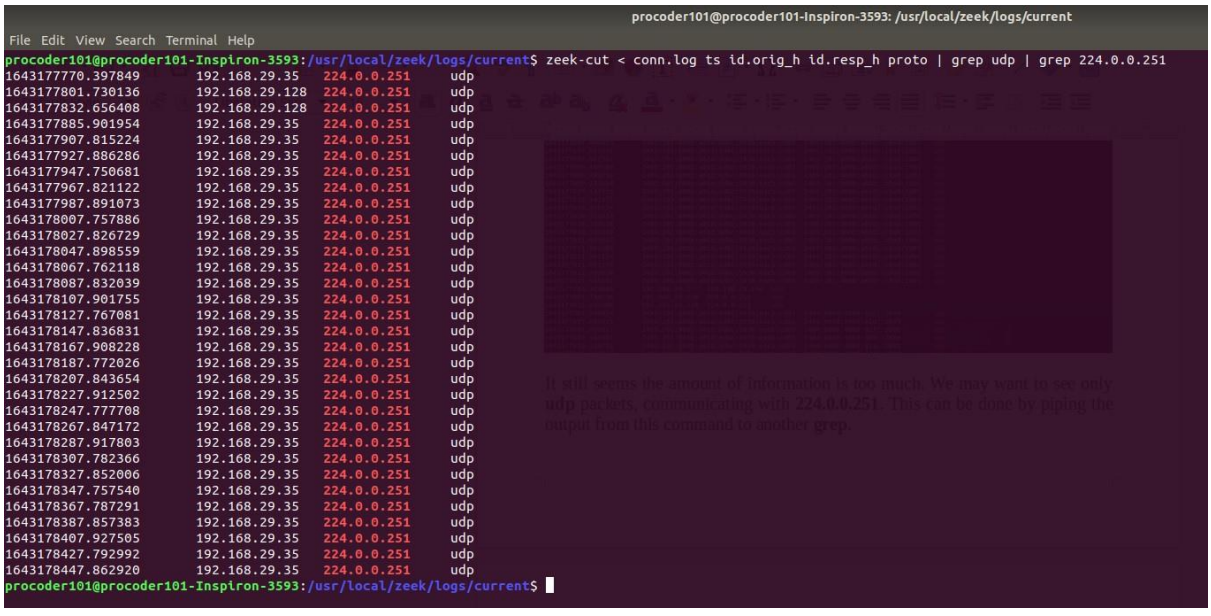
./zeek-cut < conn.log ts id.orig_h id.resp_h proto | grep udp

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



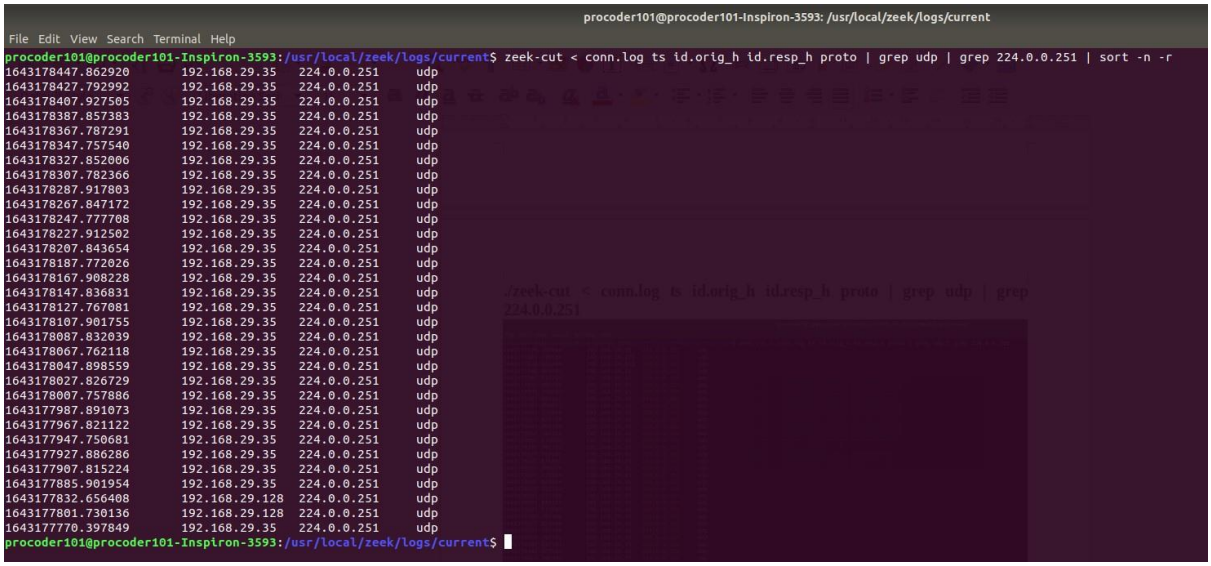
- It still seems the amount of information is too much. We may want to see only **udp** packets, communicating with **224.0.0.251**. This can be done by piping the output from this command to another **grep**.

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



- 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



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.

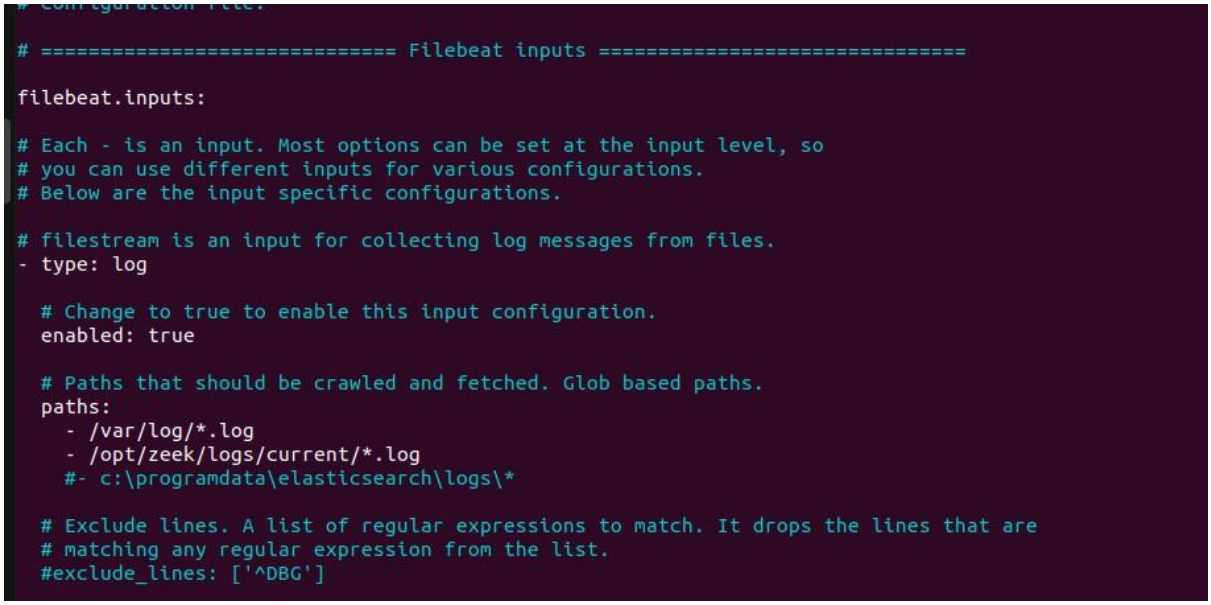
Step 1: Install Filebeat using apt

Command: - " sudo apt install filebeat "

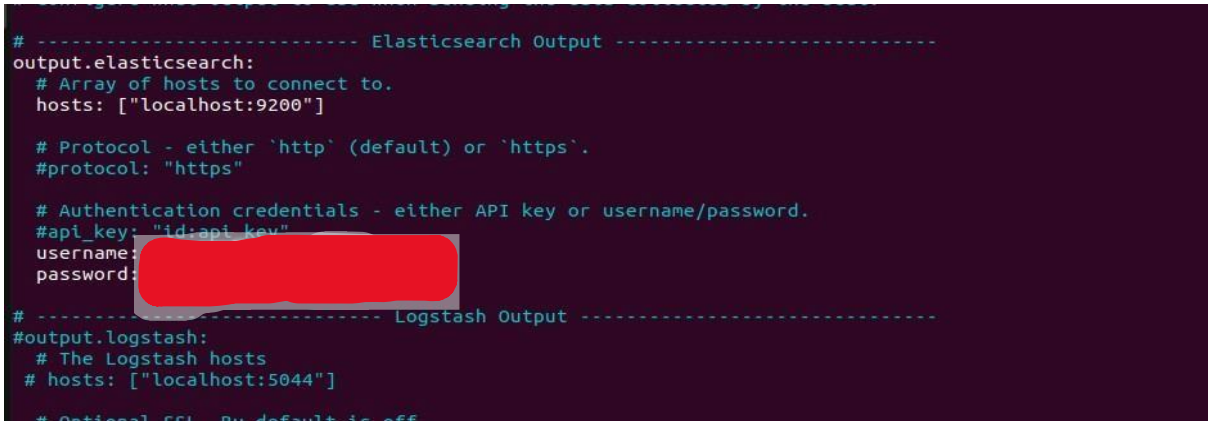
Step 2: Now we need to configure the .yml file which is present in etc/filebeat/ folder " sudo nano /etc/filebeat/filebeat.yml "

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.



- 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.



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

```
# 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" ]
  dnp3:
    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 wheather 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 repositor 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 = lLPZhZkB6oUOzzCrkLSF

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 = AN4UeQGA7HGl5iHpMla7
```

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 prerequsites 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 repositor 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: he 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:

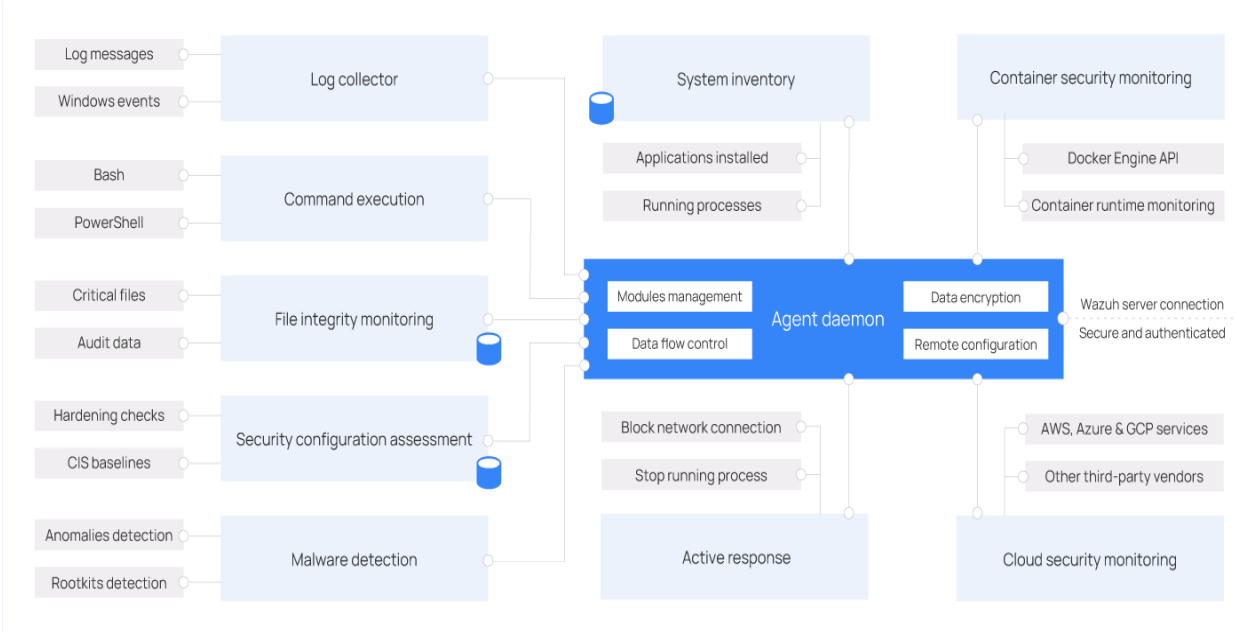
```
URL: https://<kibana_ip>

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.

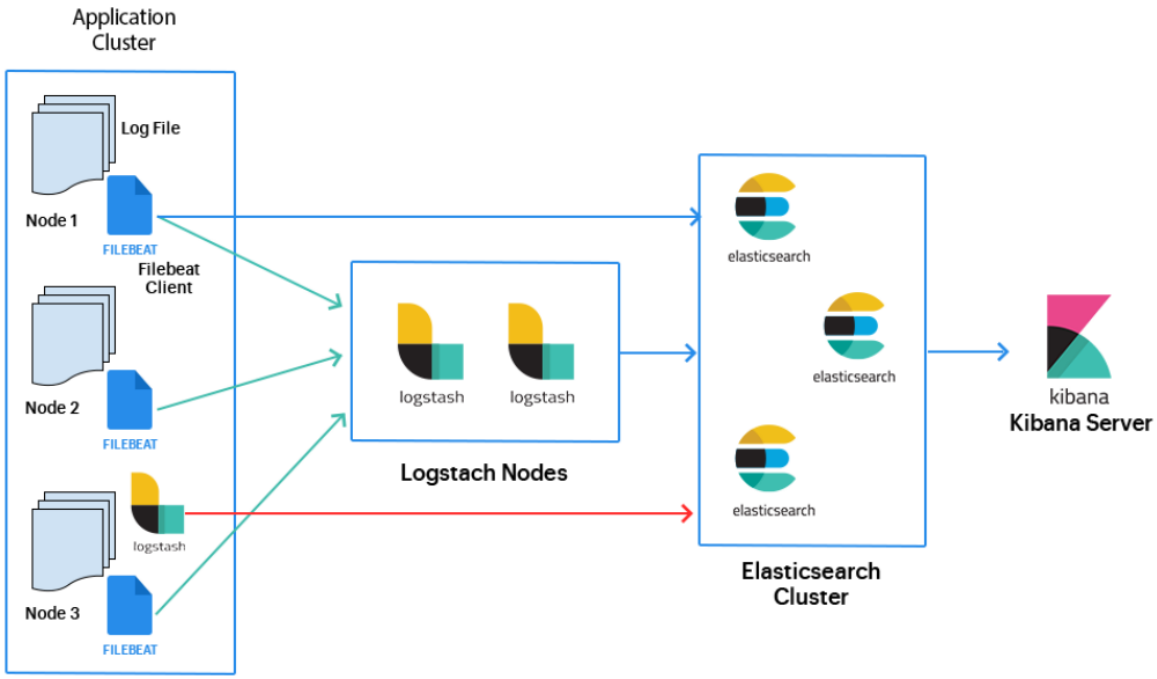
With the first access attempt, the Wazuh Kibana plugin may prompt a message that indicates that it cannot communicate with the Wazuh API. To solve this issue, edit the file /usr/share/kibana/data/wazuh/config/wazuh.yml and replace the url with the Wazuh server's address:

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

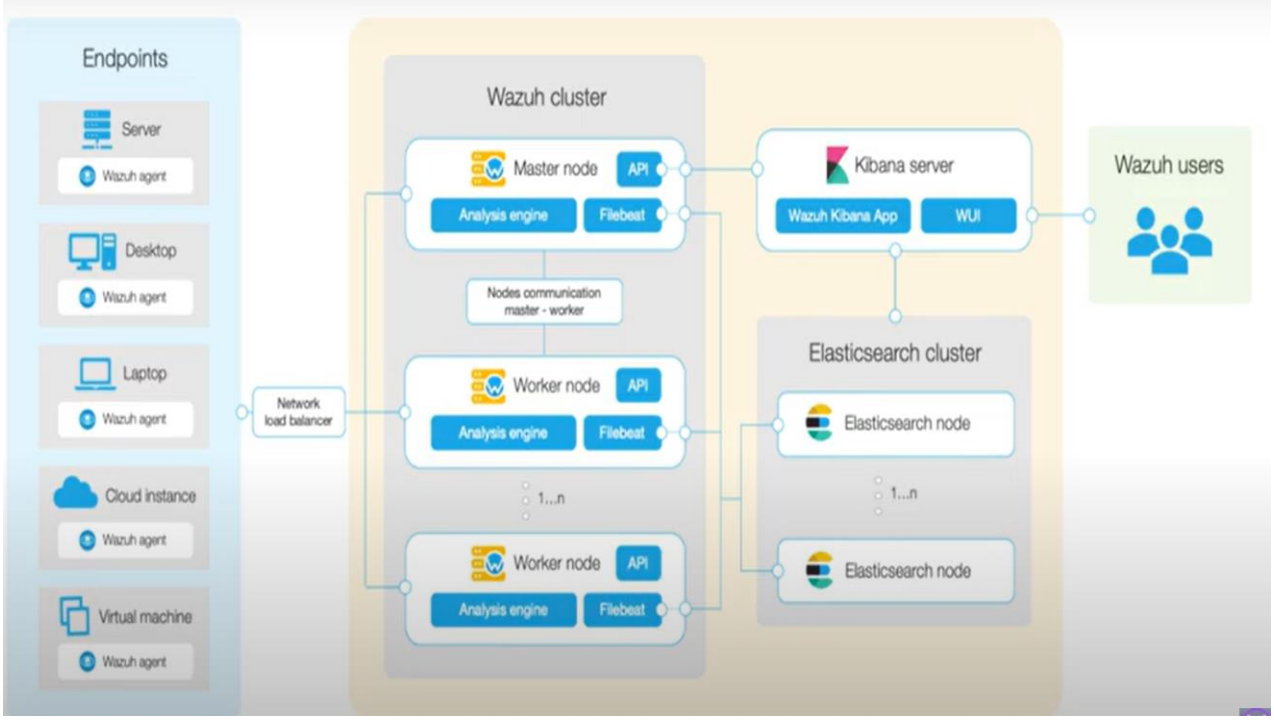
Wazuh-Agent architecture:



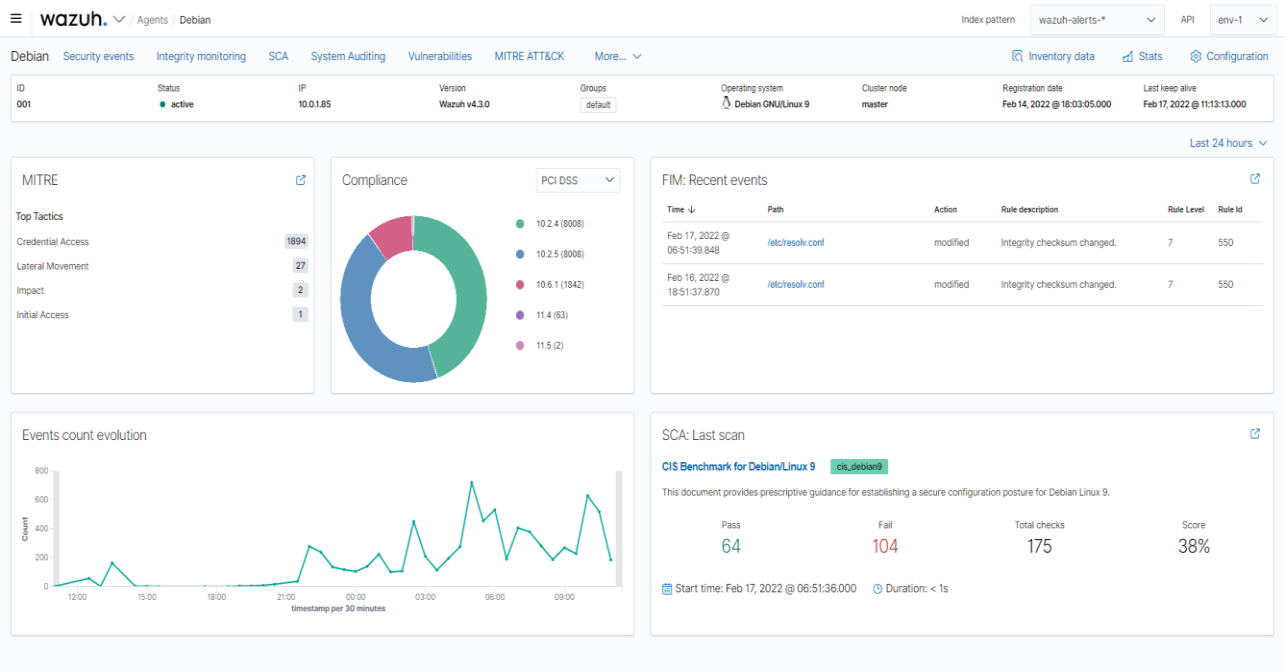
Kibana flow Diagram:



Wazuh data flow:



Wazuh-dashboard:



Acknowledgement:

The authors are grateful to Sibi Chakkaravarthy 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. Aydın, 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. Zhemelev, “Agent-based zerologon 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-7-7/230025

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