Report:

WAZUH

Strengthening Endpoint Security with Wazuh, Suricata, Virustotal, and Nmap Integration on Ubuntu.

ET.TAKI Omar

Wazuh deployment

This document guides through an installation of the Wazuh server and Elastic Stack components in an all-in-one configuration. This guide provides instructions to configure the official repositories to do the installations.

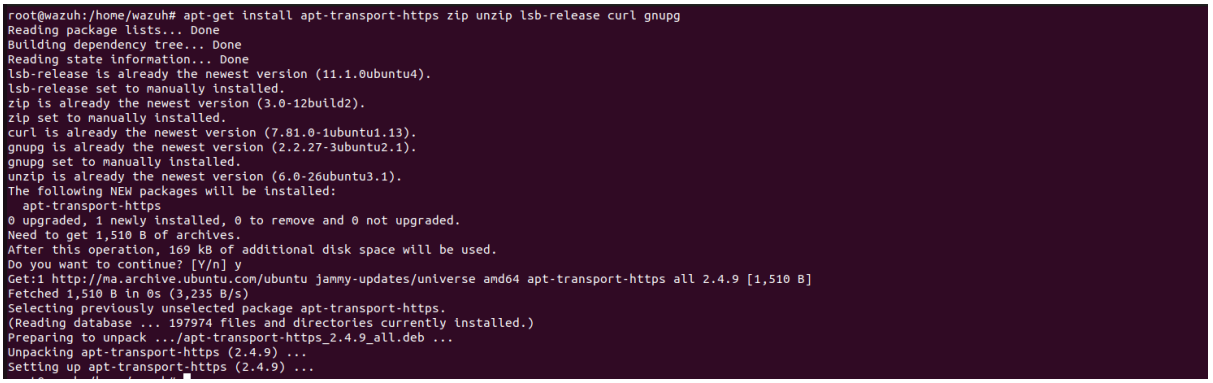
Note : You need root user privileges to run all the commands described below.

Installing prerequisites

Some extra packages are needed for the installation, such as `curl` or `unzip`.

Install all the necessary packages:

apt-get install apt-transport-https zip unzip lsb-release curl gnupg



Installing Elasticsearch

Adding the Elastic Stack repository

1. Install the GPG key:

curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/elasticsearch.gpg --import && chmod 644 /usr/share/keyrings/elasticsearch.gpg

```
root@wazuh:/home/wazuh# curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/elasticsearch.gpg --import
 && chmod 644 /usr/share/keyrings/elasticsearch.gpg
gpg: keyring '/usr/share/keyrings/elasticsearch.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key D27D666CD88E42B4: public key "Elasticsearch (Elasticsearch Signing Key) <dev_ops@elasticsearch.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

2. Add the repository:

echo "deb [signed-by=/usr/share/keyrings/elasticsearch.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | tee /etc/apt/sources.list.d/elastic-7.x.list

```
root@wazuh:/home/wazuh# echo "deb [signed-by=/usr/share/keyrings/elasticsearch.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | tee /etc/apt/sources.list.d/elas
tic-7.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main
```

3. Update the package information:

apt-get update

```
root@wazuh:/home/wazuh# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:4 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Get:5 https://artifacts.elastic.co/packages/7.x/apt stable/main i386 Packages [82.2 kB]
Hit:6 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:7 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [113 kB]
Fetched 209 kB in 8s (27.0 kB/s)
Reading package lists... Done
```

Elasticsearch installation and configuration

1. Install the Elasticsearch package:

apt-get install elasticsearch=7.17.9

```
root@wazuh:/home/wazuh# apt-get install elasticsearch=7.17.9
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 315 MB of archives.
After this operation, 527 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.9 [315 MB]
Fetched 315 MB in 7min 25s (708 kB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 197978 files and directories currently installed.)
Preparing to unpack .../elasticsearch_7.17.9_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (7.17.9) ...
Setting up elasticsearch (7.17.9) ...
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
 sudo systemctl daemon-reload
 sudo systemctl enable elasticsearch.service
### You can start elasticsearch service by executing
 sudo systemctl start elasticsearch.service
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
```

2. Download the configuration file `/etc/elasticsearch/elasticsearch.yml` as follows:

curl -so /etc/elasticsearch/elasticsearch.yml https://packages.wazuh.com/4.4/tpl/elastic-basic/elasticsearch_all_in_one.yml

```
root@wazuh:/home/wazuh# curl -so /etc/elasticsearch/elasticsearch.yml https://packages.wazuh.com/4.4/tpl/elastic-basic/elasticsearch_all_in_one.yml
```

Certificates creation and deployment

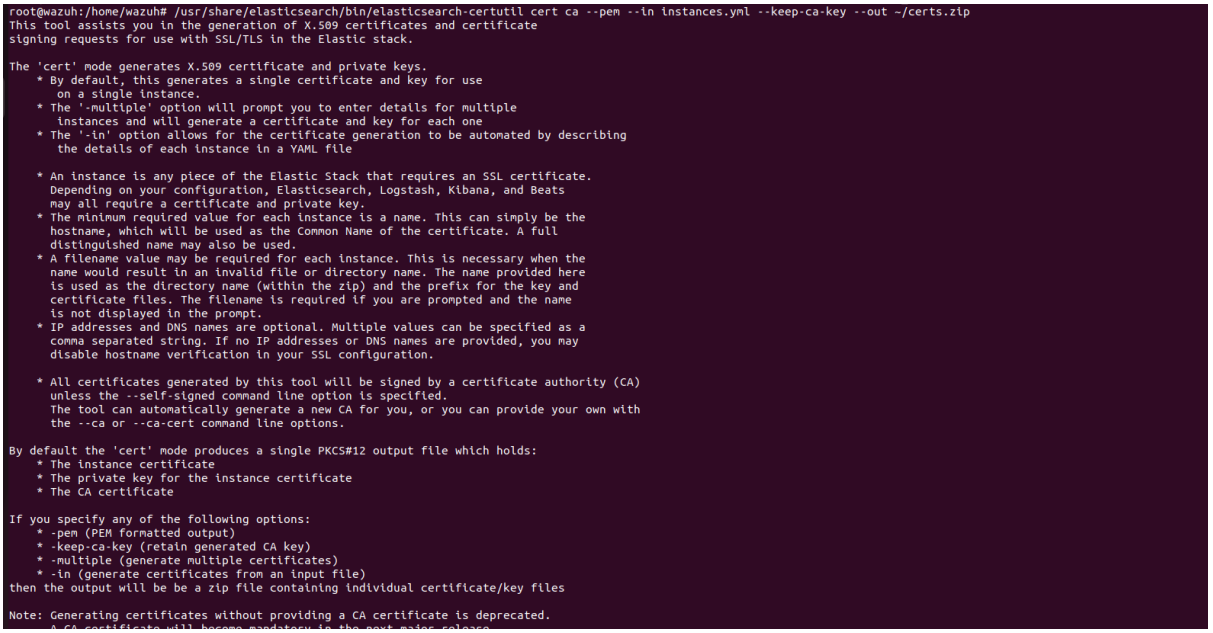
1. Download the configuration file for creating the certificates:

curl -so /usr/share/elasticsearch/instances.yml https://packages.wazuh.com/4.4/tpl/elastic-basic/instances_aio.yml

```
root@wazuh:/home/wazuh# curl -so /usr/share/elasticsearch/instances.yml https://packages.wazuh.com/4.4/tpl/elastic-basic/instances_aio.yml
```

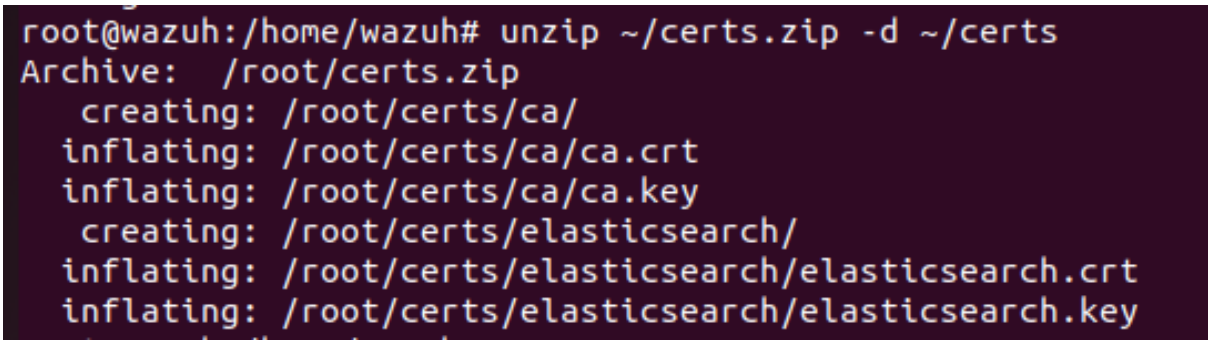
2. The certificates can be created using the elasticsearch-certutil tool:

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



3. Extract the generated `/usr/share/elasticsearch/certs.zip` file from the previous step.

unzip ~/certs.zip -d ~/certs



4. The next step is to create the directory `/etc/elasticsearch/certs`, and then copy the CA file, the certificate and the key there:

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/ ~/certs.zip

```
root@wazuh:/home/wazuh# mkdir /etc/elasticsearch/certs/ca -p
root@wazuh:/home/wazuh# cp -R ~/certs/ca/ ~/certs/elasticsearch/* /etc/elasticsearch/certs/
root@wazuh:/home/wazuh# chown -R elasticsearch: /etc/elasticsearch/certs
root@wazuh:/home/wazuh# chmod -R 500 /etc/elasticsearch/certs
root@wazuh:/home/wazuh# chmod 400 /etc/elasticsearch/certs/ca/ca.* /etc/elasticsearch/certs/elasticsearch.*
root@wazuh:/home/wazuh# rm -rf ~/certs/ ~/certs.zip
```

5. Enable and start the Elasticsearch service:

systemctl daemon-reload

systemctl enable elasticsearch

systemctl start elasticsearch

systemctl status elasticsearch

```
root@wazuh:/home/wazuh# systemctl daemon-reload
root@wazuh:/home/wazuh# systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.
root@wazuh:/home/wazuh# systemctl start elasticsearch
root@wazuh:/home/wazuh# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-07-20 11:55:25 +01; 16s ago
       Docs: https://www.elastic.co
   Main PID: 4172 (java)
      Tasks: 63 (limit: 4600)
     Memory: 2.3G
        CPU: 32.846s
     CGroup: /system.slice/elasticsearch.service
             ├─4172 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Dj
             └─4363 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

11:55:10 20 يوليوز wazuh systemd[1]: Starting Elasticsearch...
11:55:25 20 يوليوز wazuh systemd[1]: Started Elasticsearch.
lines 1-14/14 (END)
```

6. Generate credentials for all the Elastic Stack pre-built roles and users:

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

```
root@wazuh:/home/wazuh# /usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
Initiating the setup of passwords for reserved users elastic,apm_system,kibana,kibana_system,logstash_system,beats_system,remote_monitoring_user.
The passwords will be randomly generated and printed to the console.
Please confirm that you would like to continue [y/N]y


Changed password for user apm_system
PASSWORD apm_system = 4qxO20LpMHYPdFnczBCv

Changed password for user kibana_system
PASSWORD kibana_system = WU8ly9jN88gNHoqDxULE

Changed password for user kibana
PASSWORD kibana = WU8ly9jN88gNHoqDxULE

Changed password for user logstash_system
PASSWORD logstash_system = kanPWbsz38UuwuDKwMfg

Changed password for user beats_system
PASSWORD beats_system = jDCvnXOhk8TQ3Xp8CSI4

Changed password for user remote_monitoring_user
PASSWORD remote_monitoring_user = v49MQFHssFjG0V8RgJBz

Changed password for user elastic
PASSWORD elastic = RWQEP4kJbEaLDNLPIqK6
```

Save the password of the `elastic` user for further steps.
To check that the installation was made successfully, run the following command replacing `<elastic_password>` with the password generated in the previous step for `elastic` user:

curl -XGET https://localhost:9200 -u elastic:<elastic_password> -k

```
root@wazuh:/home/wazuh# curl -XGET https://localhost:9200 -u elastic:RWQEP4kJbEaLDNLPIqK6 -k
{
  "name" : "elasticsearch",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "SCGt3R1DRQ6eV3eeXsyskA",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ef48222227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Installing Wazuh server

Adding the Wazuh repository

1. Install the GPG key:

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg

```
root@wazuh:/home/wazuh# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644
/usr/share/keyrings/wazuh.gpg
gpg: keyring '/usr/share/keyrings/wazuh.gpg' created
gpg: key 96B3EE5F29111145: public key "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

2. Add the repository:

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list

```
root@wazuh:/home/wazuh# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
```

3. Update the package information:

apt-get update

```
root@wazuh:/home/wazuh# apt-get update
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://packages.wazuh.com/4.x/apt stable InRelease [17.3 kB]
Hit:6 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Get:7 https://packages.wazuh.com/4.x/apt stable/main i386 Packages [9,261 B]
Get:8 https://packages.wazuh.com/4.x/apt stable/main amd64 Packages [30.3 kB]
Fetched 56.8 kB in 1s (61.8 kB/s)
Reading package lists... Done
```

Installing the Wazuh manager

1. Install the Wazuh manager package:

apt-get install wazuh-manager

```
root@wazuh:/home/wazuh# apt-get install wazuh-manager
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  expect
The following NEW packages will be installed:
  wazuh-manager
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 171 MB of archives.
After this operation, 631 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-manager amd64 4.4.5-1 [171 MB]
Fetched 171 MB in 1min 46s (1,624 kB/s)
Selecting previously unselected package wazuh-manager.
(Reading database ... 199073 files and directories currently installed.)
Preparing to unpack .../wazuh-manager_4.4.5-1_amd64.deb ...
Unpacking wazuh-manager (4.4.5-1) ...
Setting up wazuh-manager (4.4.5-1) ...
```

2. Enable and start the Wazuh manager service:

systemctl daemon-reload
systemctl enable wazuh-manager
systemctl start wazuh-manager
systemctl status wazuh-manager

```
root@wazuh:/home/wazuh# systemctl daemon-reload
root@wazuh:/home/wazuh# systemctl enable wazuh-manager
Synchronizing state of wazuh-manager.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-manager
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-manager.service → /lib/systemd/system/wazuh-manager.service.
root@wazuh:/home/wazuh# systemctl start wazuh-manager
root@wazuh:/home/wazuh# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-07-20 12:08:02 +01; 34s ago
    Process: 48707 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 115 (limit: 4600)
     Memory: 737.2M
        CPU: 26.158s
     CGroup: /system.slice/wazuh-manager.service
             ├─48763 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─48802 /var/ossec/bin/wazuh-authd
             ├─48819 /var/ossec/bin/wazuh-db
             ├─48833 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─48836 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─48849 /var/ossec/bin/wazuh-execd
             ├─48863 /var/ossec/bin/wazuh-analysisd
             ├─48906 /var/ossec/bin/wazuh-syscheckd
             ├─48920 /var/ossec/bin/wazuh-remoted
             ├─48949 /var/ossec/bin/wazuh-logcollector
             ├─48966 /var/ossec/bin/wazuh-monitord
             └─48981 /var/ossec/bin/wazuh-modulesd

12:07:55 20 يوليوز wazuh env[48707]: Started wazuh-db...
12:07:56 20 يوليوز wazuh env[48707]: Started wazuh-execd...
12:07:58 20 يوليوز wazuh env[48707]: Started wazuh-analysisd...
12:07:59 20 يوليوز wazuh env[48707]: Started wazuh-syscheckd...
12:07:59 20 يوليوز wazuh env[48707]: Started wazuh-remoted...
12:07:59 20 يوليوز wazuh env[48707]: Started wazuh-logcollector...
12:08:00 20 يوليوز wazuh env[48707]: Started wazuh-monitord...
12:08:00 20 يوليوز wazuh env[48707]: Started wazuh-modulesd...
12:08:02 20 يوليوز wazuh env[48707]: Completed.
12:08:02 20 يوليوز wazuh systemd[1]: Started Wazuh manager.
```

Installing Filebeat

Filebeat installation and configuration

1. Install the Filebeat package:

apt-get install filebeat=7.17.9

```
root@wazuh:/home/wazuh# apt-get install filebeat=7.17.9
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 35.5 MB of archives.
After this operation, 131 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 filebeat amd64 7.17.9 [35.5 MB]
Fetched 35.5 MB in 23s (1,524 kB/s)
Selecting previously unselected package filebeat.
(Reading database ... 220639 files and directories currently installed.)
Preparing to unpack .../filebeat_7.17.9_amd64.deb ...
Unpacking filebeat (7.17.9) ...
Setting up filebeat (7.17.9) ...
```

2. Download the pre-configured Filebeat config file used to forward Wazuh alerts to Elasticsearch:

curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/4.4/tpl/elastic-basic/filebeat_all_in_one.yml

```
root@wazuh:/home/wazuh# curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/4.4/tpl/elastic-basic/filebeat_all_in_one.yml
```

3. Download the alerts template for Elasticsearch:

curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/4.4/extensions/elasticsearch/7.x/wazuh-template.json

chmod go+r /etc/filebeat/wazuh-template.json

```
root@wazuh:/home/wazuh# curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/4.4/extensions/elasticsearch/7.x/wazuh-template.json
root@wazuh:/home/wazuh# chmod go+r /etc/filebeat/wazuh-template.json
```

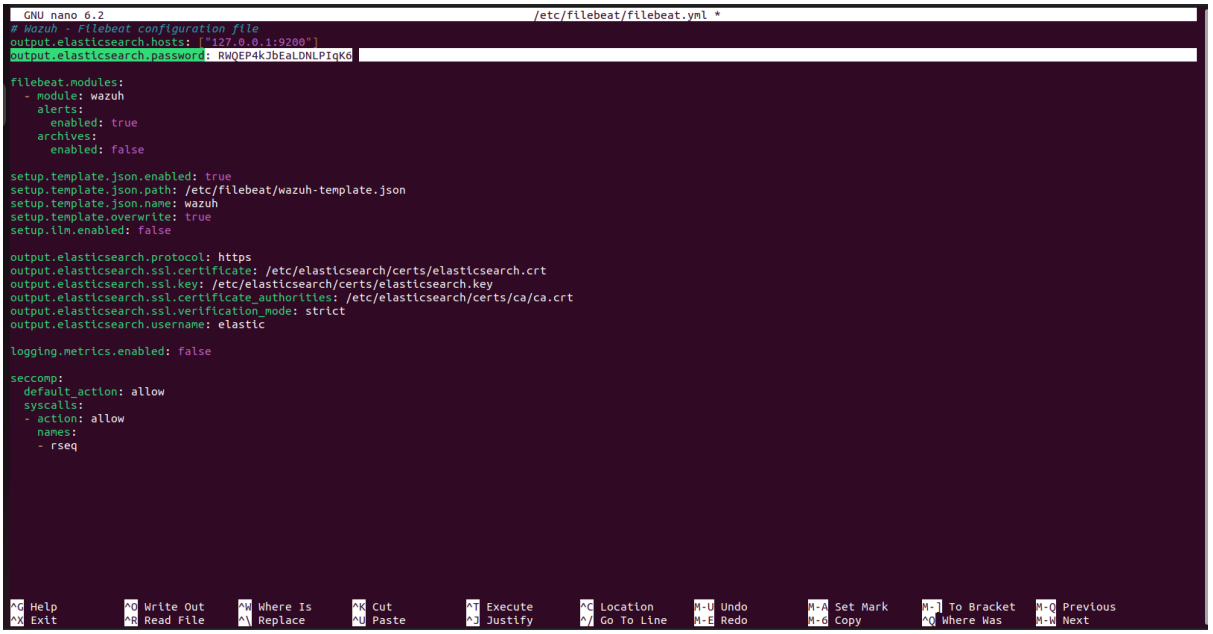
4. Download the Wazuh module for Filebeat:

curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.2.tar.gz | tar -xvz -C /usr/share/filebeat/module

```
root@wazuh:/home/wazuh# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.2.tar.gz | tar -xvz -C /usr/share/filebeat/module
wazuh/alerts/
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/manifest.yml
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/archives/
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/manifest.yml
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/module.yml
```

5. Edit the file `/etc/filebeat/filebeat.yml` and add the following line:

output.elasticsearch.password: <elasticsearch_password>



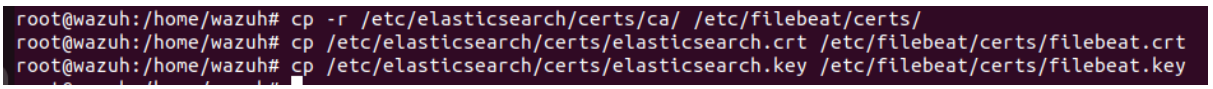
Replace elasticsearch_password with the previously generated password for elastic user.

6. Copy the certificates into /etc/filebeat/certs/

cp -r /etc/elasticsearch/certs/ca/ /etc/filebeat/certs/

cp /etc/elasticsearch/certs/elasticsearch.crt /etc/filebeat/certs/filebeat.crt

cp /etc/elasticsearch/certs/elasticsearch.key /etc/filebeat/certs/filebeat.key



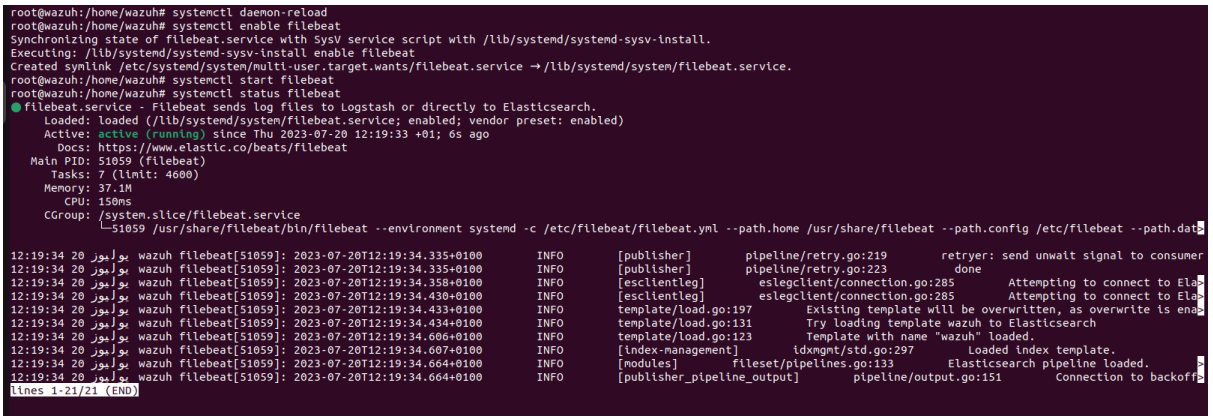
7. Enable and start the Filebeat service:

systemctl daemon-reload

systemctl enable filebeat

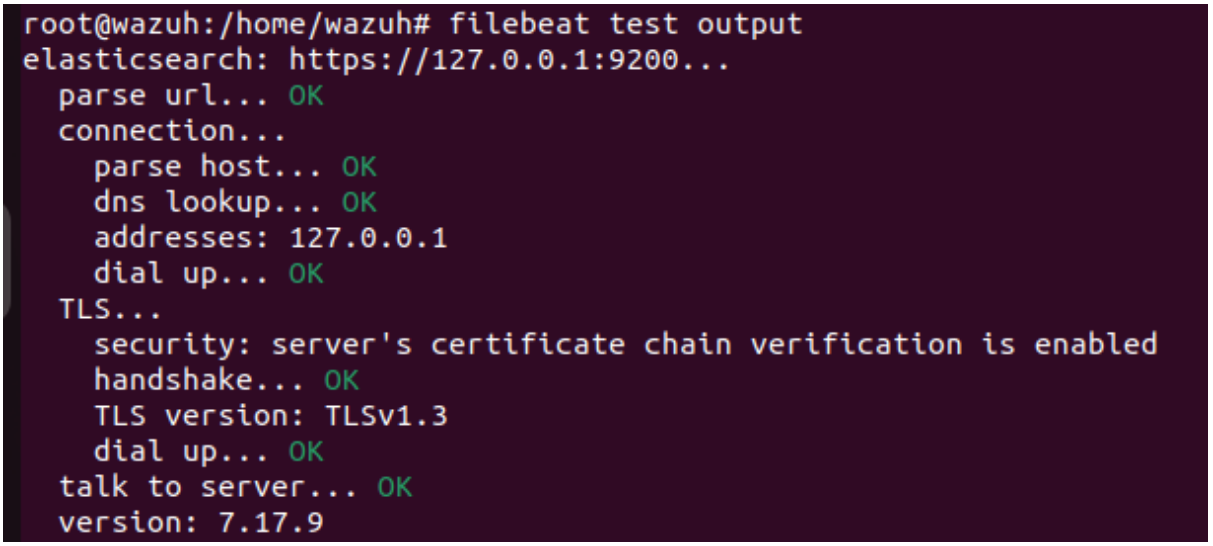
systemctl start filebeat

systemctl status filebeat



To ensure that Filebeat has been successfully installed, run the following command:

filebeat test output



Kibana installation and configuration

Install the Kibana package:

apt-get install kibana=7.17.9

```
root@wazuh:/home/wazuh# apt-get install kibana=7.17.9
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  kibana
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 272 MB of archives.
After this operation, 691 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 kibana amd64 7.17.9 [272 MB]
Fetched 272 MB in 13min 13s (343 kB/s)
Selecting previously unselected package kibana.
(Reading database ... 222715 files and directories currently installed.)
Preparing to unpack .../kibana_7.17.9_amd64.deb ...
Unpacking kibana (7.17.9) ...
Setting up kibana (7.17.9) ...
Creating kibana group... OK
Creating kibana user... OK
Created Kibana keystore in /etc/kibana/kibana.keystore
```

Copy the Elasticsearch certificates into the Kibana configuration folder:

mkdir /etc/kibana/certs/ca -p

cp -R /etc/elasticsearch/certs/ca/ /etc/kibana/certs/

cp /etc/elasticsearch/certs/elasticsearch.key /etc/kibana/certs/kibana.key

cp /etc/elasticsearch/certs/elasticsearch.crt /etc/kibana/certs/kibana.crt

chown -R kibana:kibana /etc/kibana/

chmod -R 500 /etc/kibana/certs

chmod 440 /etc/kibana/certs/ca/ca.* /etc/kibana/certs/kibana.*

```
root@wazuh:/home/wazuh# mkdir /etc/kibana/certs/ca -p
root@wazuh:/home/wazuh# cp -R /etc/elasticsearch/certs/ca/ /etc/kibana/certs/
root@wazuh:/home/wazuh# cp /etc/elasticsearch/certs/elasticsearch.key /etc/kibana/certs/kibana.key
root@wazuh:/home/wazuh# cp /etc/elasticsearch/certs/elasticsearch.crt /etc/kibana/certs/kibana.crt
root@wazuh:/home/wazuh# chown -R kibana:kibana /etc/kibana/
root@wazuh:/home/wazuh# chmod -R 500 /etc/kibana/certs
root@wazuh:/home/wazuh# chmod 440 /etc/kibana/certs/ca/ca.* /etc/kibana/certs/kibana.*
```

3. Download the Kibana configuration file:

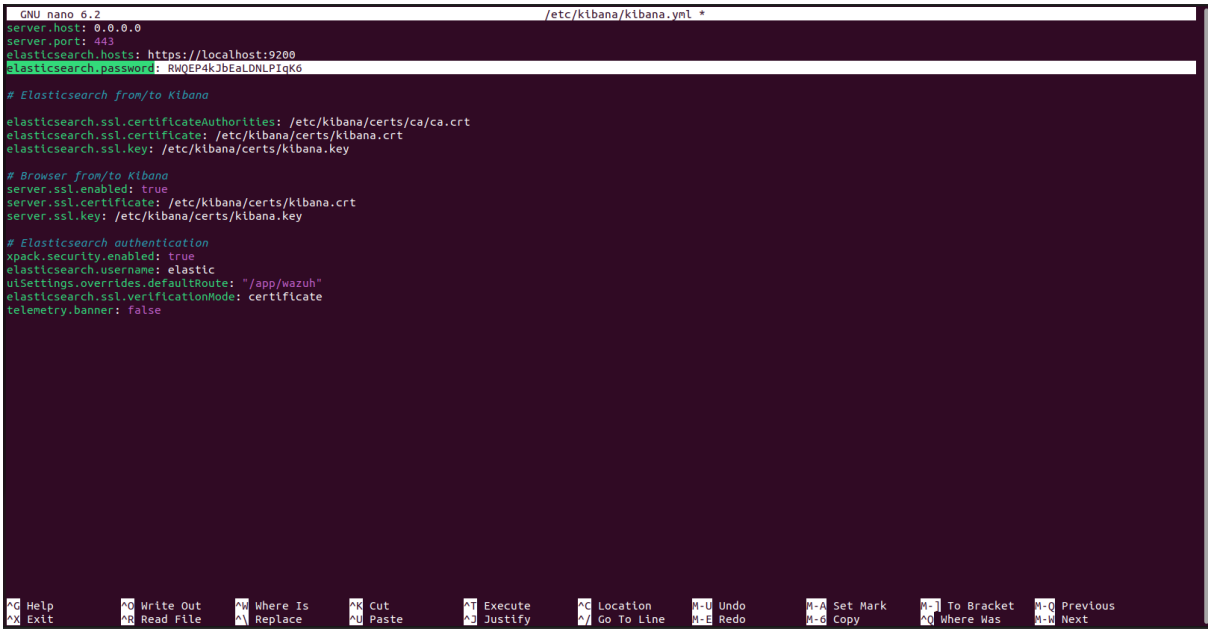
curl -so /etc/kibana/kibana.yml
https://packages.wazuh.com/4.4/tpl/elastic-basic/kibana_all_in_one.yml

```
root@wazuh:/home/wazuh# curl -so /etc/kibana/kibana.yml https://packages.wazuh.com/4.4/tpl/elastic-basic/kibana_all_in_one.yml
```

Edit the /etc/kibana/kibana.yml file:

elasticsearch.password: <elasticsearch_password>

<elasticsearch_password>: the password generated during the Elasticsearch installation and configuration for the elastic user.



4. Create the /usr/share/kibana/data directory:

mkdir /usr/share/kibana/data

chown -R kibana:kibana /usr/share/kibana

```
root@wazuh:/home/wazuh# mkdir /usr/share/kibana/data
root@wazuh:/home/wazuh# chown -R kibana:kibana /usr/share/kibana
```

5. Install the Wazuh Kibana plugin. The installation of the plugin must be done from the Kibana home directory as follows:

cd /usr/share/kibana

sudo -u kibana /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.4.5_7.17.9-1.zip

```
root@wazuh:/home/wazuh# cd /usr/share/kibana
root@wazuh:/usr/share/kibana# sudo -u kibana /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.4.5_7.17.9-1.zip
Attempting to transfer from https://packages.wazuh.com/4.x/ui/kibana/wazuh_kibana-4.4.5_7.17.9-1.zip
Transferring 36505918 bytes....................
Transfer complete
Retrieving metadata from plugin archive
Extracting plugin archive
Extraction complete
Plugin installation complete
```

6. Link Kibana's socket to privileged port 443:

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

```
root@wazuh:/usr/share/kibana# setcap 'cap_net_bind_service=+ep' /usr/share/kibana/node/bin/node
```

7. Enable and start the Kibana service:

systemctl daemon-reload

systemctl enable kibana

systemctl start kibana

systemctl status kibana

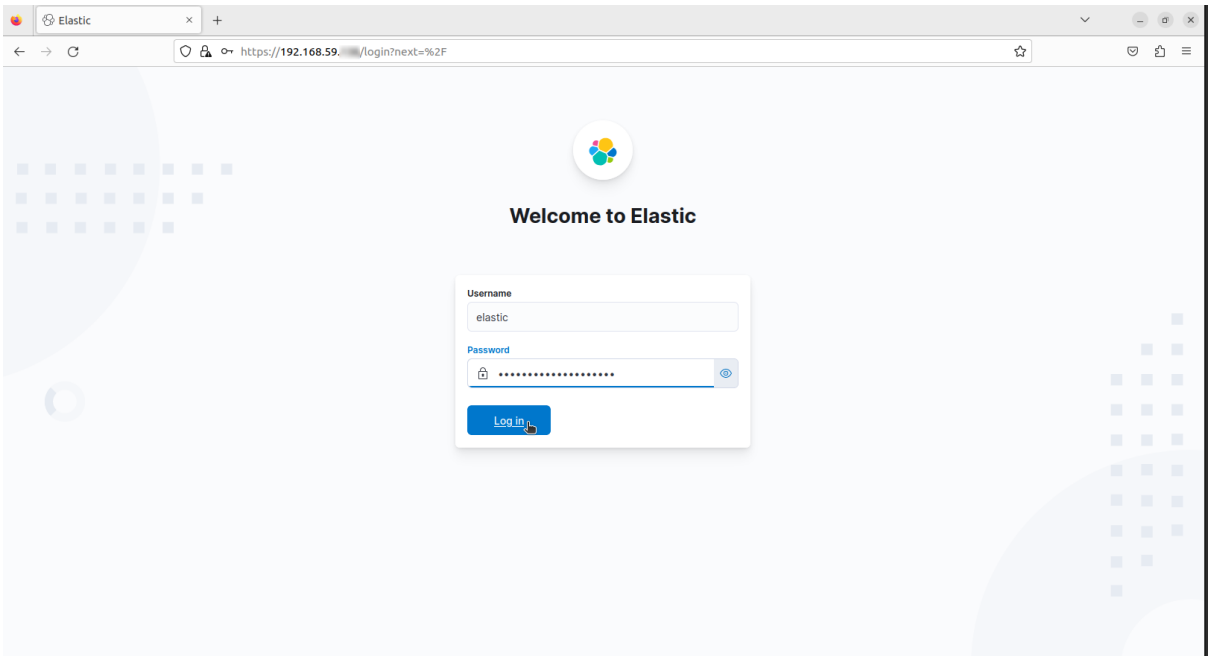
```
root@wazuh:/usr/share/kibana# systemctl daemon-reload
root@wazuh:/usr/share/kibana# systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /etc/systemd/system/kibana.service.
root@wazuh:/usr/share/kibana# systemctl start kibana
root@wazuh:/usr/share/kibana# systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-07-20 12:48:46 +01; 4s ago
       Docs: https://www.elastic.co
   Main PID: 51529 (node)
      Tasks: 11 (limit: 4600)
     Memory: 217.5M
        CPU: 4.573s
     CGroup: /system.slice/kibana.service
             └─51529 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --logging.dest=/var/log/kibana/kibana.log --pid.file=/run/kibana/kibana.pid "-

12:48:46 20 يوليو wazuh systemd[1]: Started Kibana.
lines 1-12/12 (END)
```

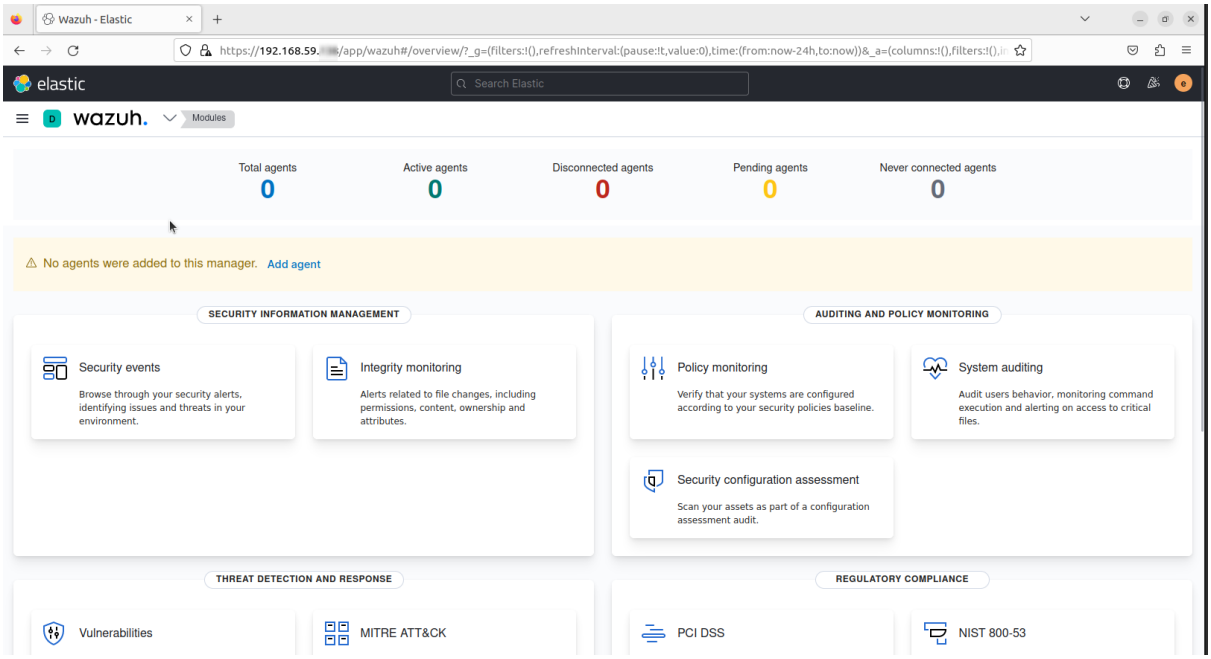
8. Access the web interface using the password generated during the Elasticsearch installation process:

URL: https://<wazuh_server_ip>

user: elastic

password: <PASSWORD_elastic>





Disabling repositories

This installation guide describes how to install and configure Wazuh and Elastic Stack by first configuring their repositories.

With each new release of Wazuh or Elastic Stack, the development team at Wazuh thoroughly tests the compatibility of each component and performs necessary adjustments before releasing a new Wazuh Kibana plugin.

We recommend disabling the repositories so that the individual packages will not be updated unintentionally, which could potentially lead to having a version of the Elastic Stack for which the Wazuh integration has not been released yet.

```
# sed -i "s/^deb/#deb/" /etc/apt/sources.list.d/wazuh.list
# sed -i "s/^deb/#deb/" /etc/apt/sources.list.d/elastic-7.x.list
# apt-get update
```

```
root@wazuh:/usr/share/kibana# sed -i "s/^deb/#deb/" /etc/apt/sources.list.d/wazuh.list
root@wazuh:/usr/share/kibana# sed -i "s/^deb/#deb/" /etc/apt/sources.list.d/elastic-7.x.list
root@wazuh:/usr/share/kibana# apt-get update
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [15.8 kB]
Get:6 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [946 kB]
Get:7 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [204 kB]
Fetched 1,502 kB in 2s (612 kB/s)
Reading package lists... Done
```

Deploying Wazuh agents on Linux endpoints

The agent runs on the host you want to monitor and communicates with the Wazuh server, sending data in near real-time through an encrypted and authenticated channel.

Note: You need root user privileges to run all the commands described below.

Add the Wazuh repository

Add the Wazuh repository to download the official packages.

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg

```
root@agent-virtual-machine:/home/agent# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --impor
t && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 96B3EE5F29111145: public key "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list

```
root@agent-virtual-machine:/home/agent# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazu
h.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
```

apt-get update

```
root@agent-virtual-machine:/home/agent# apt-get update
Get:1 https://packages.wazuh.com/4.x/apt stable InRelease [17.3 kB]
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:3 https://packages.wazuh.com/4.x/apt stable/main amd64 Packages [32.1 kB]
Get:4 https://packages.wazuh.com/4.x/apt stable/main i386 Packages [9,695 B]
Hit:5 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Hit:6 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:7 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 59.1 kB in 1s (53.3 kB/s)
Reading package lists... Done
```

Deploy a Wazuh agent

To deploy the Wazuh agent on your endpoint, select your package manager and edit the `WAZUH_MANAGER` variable to contain your Wazuh manager IP address or hostname.

WAZUH_MANAGER="192.168.59.140" apt-get install wazuh-agent

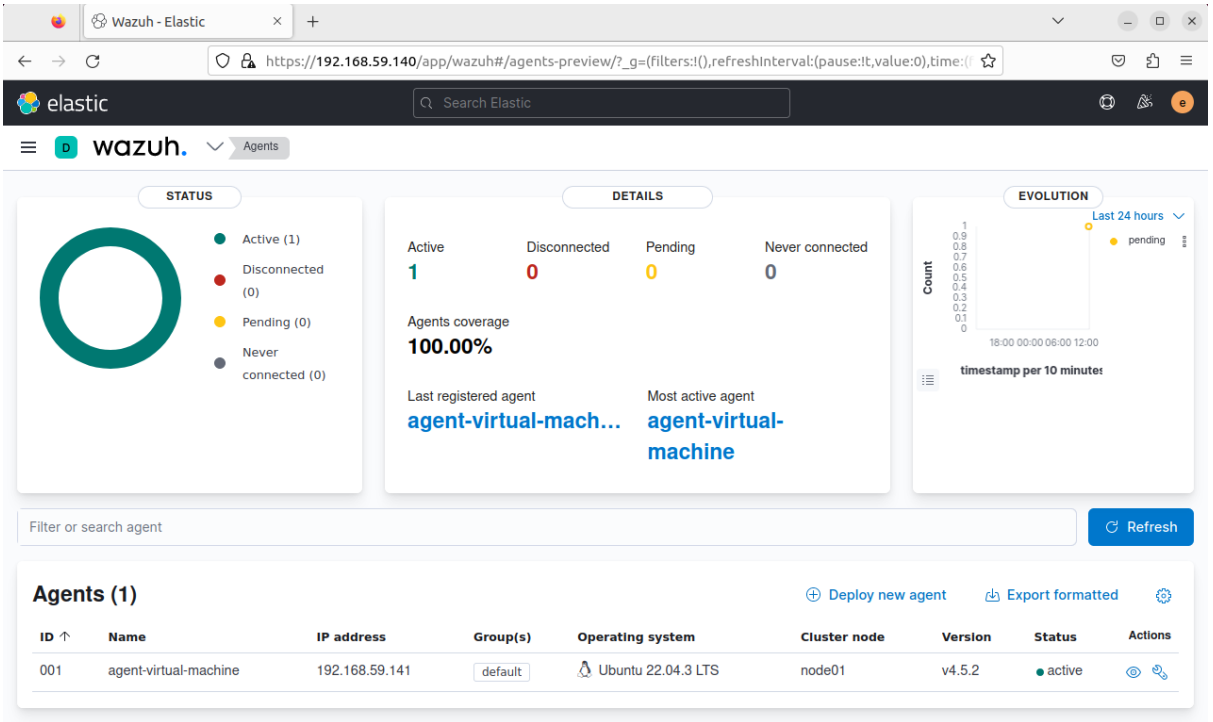
```
root@agent-virtual-machine:/home/agent# WAZUH_MANAGER="192.168.59.140" apt-get install wazuh-agent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  wazuh-agent
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 8,943 kB of archives.
After this operation, 30.0 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-agent amd64 4.5.2-1 [8,943 kB]
Fetched 8,943 kB in 3s (2,745 kB/s)
Preconfiguring packages ...
Selecting previously unselected package wazuh-agent.
(Reading database ... 198405 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.5.2-1_amd64.deb ...
Unpacking wazuh-agent (4.5.2-1) ...
Setting up wazuh-agent (4.5.2-1) ...
```

Enable and start the Wazuh agent service.

systemctl daemon-reload
systemctl enable wazuh-agent
systemctl start wazuh-agent
systemctl status wazuh-agent

```
root@agent-virtual-machine:/home/agent# systemctl daemon-reload
systemctl enable wazuh-agent
systemctl start wazuh-agent
Synchronizing state of wazuh-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/system/wazuh-agent.service.
root@agent-virtual-machine:/home/agent# systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; vendor p
     Active: active (running) since Thu 2023-09-21 12:27:40 +01; 7s ago
    Process: 3747 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (co
      Tasks: 28 (limit: 4556)
     Memory: 18.7M
        CPU: 1.435s
     CGroup: /system.slice/wazuh-agent.service
             ├─3769 /var/ossec/bin/wazuh-execd
             ├─3777 /var/ossec/bin/wazuh-agentd
             ├─3790 /var/ossec/bin/wazuh-syscheckd
             ├─3802 /var/ossec/bin/wazuh-logcollector
             └─3819 /var/ossec/bin/wazuh-modulesd
```

the Wazuh dashboard



Integrate Suricata with Wazuh for Log Processing

Follow through this tutorial to learn how to integrate Suricata with Wazuh for log processing. With its ability to write its logs in YAML and JSON formats, Suricata can be integrated with other tools such as SIEMs, Splunk, Logstash/Elasticsearch, Kibana for further logs processing and visualization. In this tutorial, we will see how you can easily integrated it with Wazuh, an open-source *threat detection, security monitoring, incident response and regulatory compliance* tool to process the Suricata generated alerts for better monitoring and visualization network traffic.

How to Integrate Suricata with Wazuh for Log Processing

To integrate Suricata with Wazuh for log processing;

- Install and Setup Wazuh Server
- Install Wazuh Agents
- Install and Setup Suricata on Ubuntu

Install and Setup Suricata on Ubuntu

Install Suricata from Source On Ubuntu

Installation Suricata from the Source on Ubuntu is the surest way to get the latest and stable version of Suricata up and running.

To install Suricata from the source, first install all the required dependencies installed.

```
# sudo apt -y install libpcre3 libpcre3-dbg libpcre3-dev \
build-essential autoconf automake libtool libpcap-dev \
libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev \
libcap-ng-dev libcap-ng0 make libmagic-dev \
libjansson-dev libjansson4 pkg-config libnspr4-dev \
libnss3-dev liblz4-dev rustc cargo python3-pip python3-distutils
```

```
root@agent-virtual-machine:/home/agent# apt -y install libpcre3 libpcre3-dbg libpcre3-dev \
build-essential autoconf automake libtool libpcap-dev \
libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev \
libcap-ng-dev libcap-ng0 make libmagic-dev \
libjansson-dev libjansson4 pkg-config libnspr4-dev \
libnss3-dev liblz4-dev rustc cargo python3-pip python3-distutils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libcap-ng0 is already the newest version (0.7.9-2.2build3).
libcap-ng0 set to manually installed.
libjansson4 is already the newest version (2.13.1-1.1build3).
libjansson4 set to manually installed.
libyaml-0-2 is already the newest version (0.2.2-1build2).
libyaml-0-2 set to manually installed.
libpcre3 is already the newest version (2:8.39-13ubuntu0.22.04.1).
libpcre3 set to manually installed.
zlib1g is already the newest version (1:1.2.11.dfsg-2ubuntu9.2).
zlib1g set to manually installed.
The following additional packages will be installed:
  autotools-dev binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++ g++-11 gcc gcc-11 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdbus-1-dev libdpkg-perl libexpat1-dev
  libfakeroot libfile-fcntllock-perl libgcc-11-dev libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libltdl-dev libnet1 libnsl-dev libpcap0.8-dev libpcre16-3
  libpcre32-3 libpcrecpp0v5 libpython3-dev libpython3.10-dev libquadmath0 libsigsegv2 libssh2-1 libstd-rust-1.66 libstd-rust-dev libstdc++-11-dev libtirpc-dev libtsan0
  libubsan1 linux-libc-dev lto-disabled-list m4 manpages-dev python3-dev python3-setuptools python3-wheel python3.10-dev rpcsvc-proto
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc gettext binutils-doc cargo-doc debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib flex bison gcc-doc
  gcc-11-multilib gcc-11-locales apache2 | lighttpd | httpd glibc-doc git bzr libtool-doc libstdc++-11-doc gfortran | fortran95-compiler gcj-jdk libyaml-doc m4-doc make-doc
  python-setuptools-doc llvm-15 lld-15 clang-15
The following NEW packages will be installed:
  autoconf automake autotools-dev binutils binutils-common binutils-x86-64-linux-gnu build-essential cargo dpkg-dev fakeroot g++ g++-11 gcc gcc-11 javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcap-ng-dev libcc1-0 libcrypt-dev
  libctf-nobfd0 libctf0 libdbus-1-dev libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-11-dev libitm1 libjansson-dev libjs-jquery libjs-sphinxdoc
  libjs-underscore liblsan0 libltdl-dev liblz4-dev libmagic-dev libnet1 libnet1-dev libnsl-dev libnspr4-dev libnss3-dev libpcap-dev libpcap0.8-dev libpcre16-3 libpcre3-dbg
  libpcre3-dev libpcre32-3 libpcrecpp0v5 libpython3-dev libpython3.10-dev libquadmath0 libsigsegv2 libssh2-1 libstd-rust-1.66 libstd-rust-dev libstdc++-11-dev libtirpc-dev
  libtool libtsan0 libubsan1 libyaml-dev linux-libc-dev lto-disabled-list m4 make manpages-dev pkg-config python3-dev python3-distutils python3-pip python3-setuptools
  python3-wheel python3.10-dev rpcsvc-proto rustc zlib1g-dev
0 upgraded, 83 newly installed, 0 to remove and 0 not upgraded.
Need to get 139 MB of archives.
After this operation, 539 MB of additional disk space will be used.
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libsigsegv2 amd64 2.13-1ubuntu3 [14.6 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 m4 amd64 1.4.18-5ubuntu2 [199 kB]
```

Suricata function as an IDS out of the box. If you need to include the IPS funtionality, install the following libraries.
sudo apt -y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0

```
root@agent-virtual-machine:/home/agent# sudo apt -y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libnfnetlink0 is already the newest version (1.0.1-3build3).
libnfnetlink0 set to manually installed.
Suggested packages:
  libnetfilter-queue-doc
The following NEW packages will be installed:
  libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 36.8 kB of archives.
After this operation, 160 kB of additional disk space will be used.
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy/universe amd64 libnetfilter-queue1 amd64 1.0.5-2 [14.4 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libnfnetlink-dev amd64 1.0.1-3build3 [6,418 B]
Get:3 http://ma.archive.ubuntu.com/ubuntu jammy/universe amd64 libnetfilter-queue-dev amd64 1.0.5-2 [16.0 kB]
Fetched 36.8 kB in 1s (48.7 kB/s)
Selecting previously unselected package libnetfilter-queue1:amd64.
(Reading database ... 206958 files and directories currently installed.)
Preparing to unpack .../libnetfilter-queue1_1.0.5-2_amd64.deb ...
Unpacking libnetfilter-queue1:amd64 (1.0.5-2) ...
Selecting previously unselected package libnfnetlink-dev.
Preparing to unpack .../libnfnetlink-dev_1.0.1-3build3_amd64.deb ...
Unpacking libnfnetlink-dev (1.0.1-3build3) ...
Selecting previously unselected package libnetfilter-queue-dev:amd64.
Preparing to unpack .../libnetfilter-queue-dev_1.0.5-2_amd64.deb ...
Unpacking libnetfilter-queue-dev:amd64 (1.0.5-2) ...
Setting up libnfnetlink-dev (1.0.1-3build3) ...
Setting up libnetfilter-queue1:amd64 (1.0.5-2) ...
Setting up libnetfilter-queue-dev:amd64 (1.0.5-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.3) ...
```

Next, download the [latest and stable Suricata tarball](). You can simply download as shown below;

#wget https://www.openinfosecfoundation.org/download/suricata-6.0.5.tar.gz

```
root@agent-virtual-machine:/home/agent# wget https://www.openinfosecfoundation.org/download/suricata-6.0.5.tar.gz
--2023-09-21 15:39:43--  https://www.openinfosecfoundation.org/download/suricata-6.0.5.tar.gz
Resolving www.openinfosecfoundation.org (www.openinfosecfoundation.org)... 52.14.249.179, 2600:1f16:db2:4f00:da9d:37d6:e8b9:9802
Connecting to www.openinfosecfoundation.org (www.openinfosecfoundation.org)|52.14.249.179|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32605145 (31M) [application/x-gzip]
Saving to: ‘suricata-6.0.5.tar.gz’

suricata-6.0.5.tar.gz                      100%[=========================================================================================>]  31.09M  2.16MB/s    in 15s

2023-09-21 15:39:59 (2.03 MB/s) - ‘suricata-6.0.5.tar.gz’ saved [32605145/32605145]
```

Once the download is complete, extract the tarball.

tar xzf suricata-6.0.5.tar.gz

Navigate to Suricata tarball extract directory to configure Suricata engine for compilation. This ensures that Suricata is build with IPS capabilities.

cd suricata-6.0.5
./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --localstatedir=/var

```
root@agent-virtual-machine:/home/agent# tar xzf suricata-6.0.5.tar.gz
root@agent-virtual-machine:/home/agent# cd suricata-6.0.5
./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --localstatedir=/var
checking whether make supports nested variables... yes
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether UID '0' is supported by ustar format... yes
checking whether GID '0' is supported by ustar format... yes
checking how to create a ustar tar archive... gnutar
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking how to print strings... printf
checking whether make supports the include directive... yes (GNU style)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
```

Compile and install the Suricata engine

make install-full

```
root@agent-virtual-machine:/home/agent/suricata-6.0.5# make install-full
make install
make[1]: Entering directory '/home/agent/suricata-6.0.5'
Making install in libhtp
make[2]: Entering directory '/home/agent/suricata-6.0.5/libhtp'
Making install in htp
make[3]: Entering directory '/home/agent/suricata-6.0.5/libhtp/htp'
Making install in lzma
make[4]: Entering directory '/home/agent/suricata-6.0.5/libhtp/htp/lzma'
/bin/bash ../../libtool  --tag=CC   --mode=compile gcc -DHAVE_CONFIG_H -I. -I../..   -O2 -I../.. -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-in
itializers -Wshadow -Wpointer-arith -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat
-security -fPIC -MT LzFind.lo -MD -MP -MF .deps/LzFind.Tpo -c -o LzFind.lo LzFind.c
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../.. -O2 -I../.. -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-initializers -Wshadow -Wpointer-arit
h -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC -MT LzFind.lo -MD
-MP -MF .deps/LzFind.Tpo -c LzFind.c  -fPIC -DPIC -o .libs/LzFind.o
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../.. -O2 -I../.. -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-initializers -Wshadow -Wpointer-arit
h -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC -MT LzFind.lo -MD
-MP -MF .deps/LzFind.Tpo -c LzFind.c -o LzFind.o >/dev/null 2>&1
mv -f .deps/LzFind.Tpo .deps/LzFind.Plo
/bin/bash ../../libtool  --tag=CC   --mode=compile gcc -DHAVE_CONFIG_H -I. -I../..   -O2 -I../.. -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-in
itializers -Wshadow -Wpointer-arith -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat
-security -fPIC -MT LzmaDec.lo -MD -MP -MF .deps/LzmaDec.Tpo -c -o LzmaDec.lo LzmaDec.c
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../.. -O2 -I../.. -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-initializers -Wshadow -Wpointer-arit
h -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC -MT LzmaDec.lo -MD
 -MP -MF .deps/LzmaDec.Tpo -c LzmaDec.c  -fPIC -DPIC -o .libs/LzmaDec.o
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../.. -O2 -I../.. -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-initializers -Wshadow -Wpointer-arit
h -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC -MT LzmaDec.lo -MD
 -MP -MF .deps/LzmaDec.Tpo -c LzmaDec.c -o LzmaDec.o >/dev/null 2>&1
mv -f .deps/LzmaDec.Tpo .deps/LzmaDec.Plo
/bin/bash ../../libtool  --tag=CC   --mode=link gcc -I../.. -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-initializers -Wshadow -Wpointer-arith -
Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC   -o liblzma-c.la
LzFind.lo LzmaDec.lo  -lz
libtool: link: ar cru .libs/liblzma-c.a .libs/LzFind.o .libs/LzmaDec.o
ar: `u' modifier ignored since `D' is the default (see `U')
libtool: link: ranlib .libs/liblzma-c.a
libtool: link: ( cd ".libs" && rm -f "liblzma-c.la" && ln -s "../liblzma-c.la" "liblzma-c.la" )
make[5]: Entering directory '/home/agent/suricata-6.0.5/libhtp/htp/lzma'
make[5]: Nothing to be done for 'install-exec-am'.
 /usr/bin/mkdir -p '/usr/include/htp/lzma'
 /usr/bin/install -c -m 644 LzmaDec.h 7zTypes.h '/usr/include/htp/lzma'
make[5]: Leaving directory '/home/agent/suricata-6.0.5/libhtp/htp/lzma'
make[4]: Leaving directory '/home/agent/suricata-6.0.5/libhtp/htp/lzma'
make[4]: Entering directory '/home/agent/suricata-6.0.5/libhtp/htp'
/bin/bash ../libtool  --tag=CC   --mode=compile gcc -DHAVE_CONFIG_H -I. -I..   -O2 -I.. -I../htp -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-in
itializers -Wshadow -Wpointer-arith -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat
-security -fPIC -MT bstr.lo -MD -MP -MF .deps/bstr.Tpo -c -o bstr.lo bstr.c
```

Configure Suricata on Ubuntu

At the end of installation, you will have Suricata rules under `/etc/suricata/rules/` and the main configuration file under `/etc/suricata/suricata.yaml`.

The default Suricata configuration file commented well enough to provide a clear understanding of what every setting is for.

To begin with, you need to configure Suricata to differentiate between your internal network to be proctected and external network. This can be done by defining the correct values for the `HOME_NET` and `EXTERNAL_NET` variables respectively under the address groups.

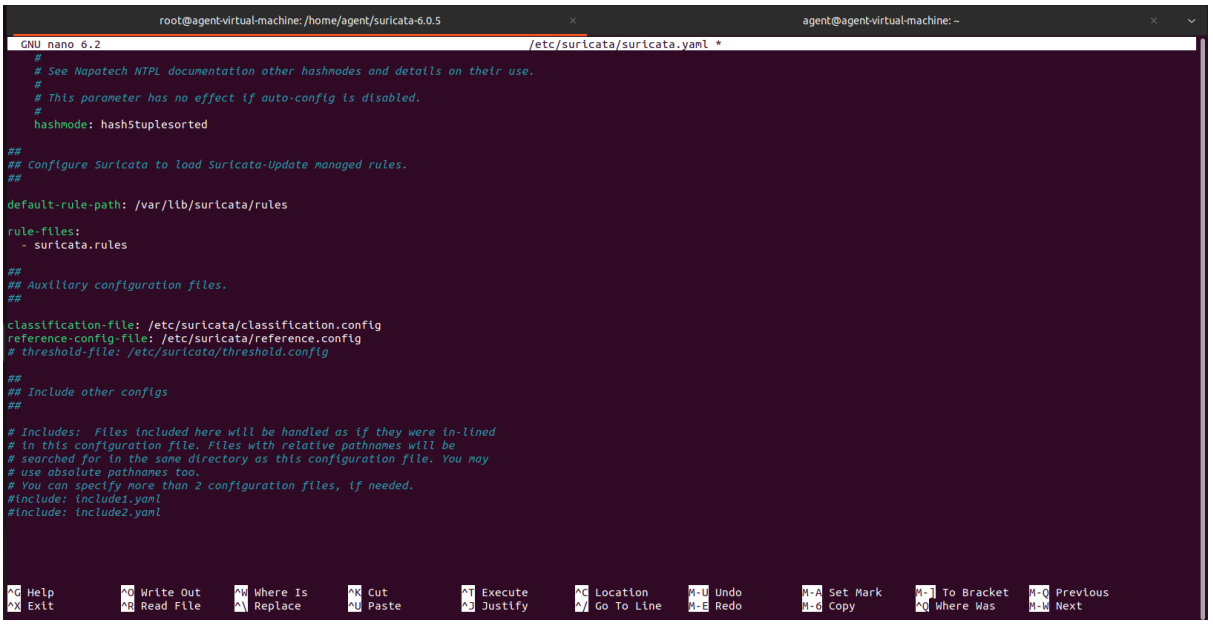
nano /etc/suricata/suricata.yaml

In my case, am using the IP address, `192.168.59.0/24`, as my home network. The external networks are set to any that doesn't match the home networks.

You can define multiple networks.

Also, define the interface on which Suricata will use to inspect the traffic.

So get your interfaces using the ip command and determine which one to configure Suricata to use.

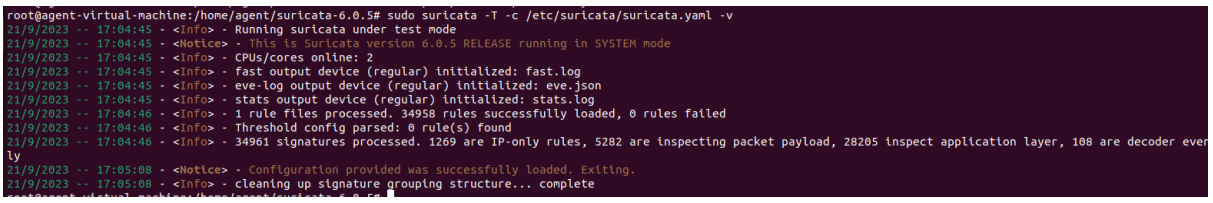


Running Suricata on Ubuntu

You can now start and enable Suricata service to run on system boot.

However, always before you start Suricata, run the configuration check;

sudo suricata -T -c /etc/suricata/suricata.yaml -v



In case of any error, fix it before you can start Suricata

If there is no error, then start Suricata;

sudo systemctl enable --now suricata

You can check the status;

sudo systemctl status suricata

Configure Suricata Logging

By default, Suricata logs alerts to two different files;

- `fast.log`: which contains line based alerts log
- `eve.json`: which stores the event logs in JSON format

Configure Wazuh Agent to Collect Suricata Logs

In order to be able to integrate Suricata with Wazuh for log processing, you need to configure Wazuh agent to read the Suricata EVE logs.

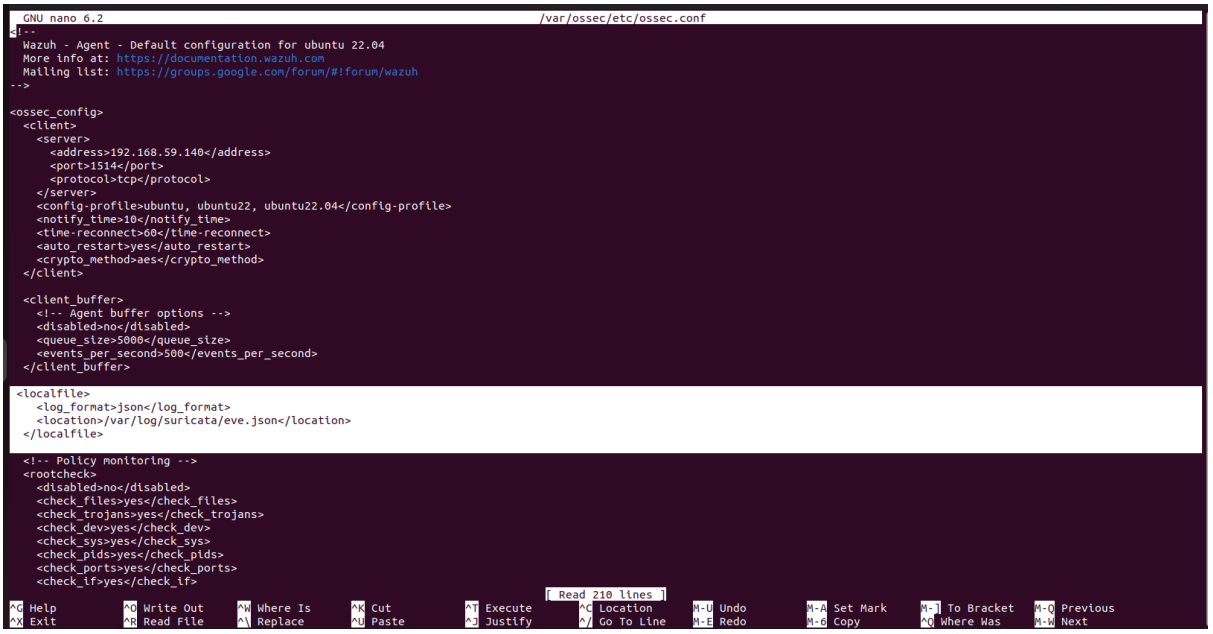
The Suricata EVE log file is usually `/var/log/suricata/eve.json` by default.

Thus, open the Wazuh agent configuration file for editing;

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

Add the lines below, just before the last line, `</ossec_config>`.

```
<localfile>
  <log_format>json</log_format>
  <location>/var/log/suricata/eve.json</location>
</localfile>
```



Save and exit the file.

Restart Wazuh agent service

Before you can restart Wazuh agent, run the command below to check if any configuration syntax error;

sudo /var/ossec/bin/wazuh-syscheckd -t

exit status should be 0 if no error. Otherwise, you will see a message about the error.

sudo systemctl restart wazuh-agent

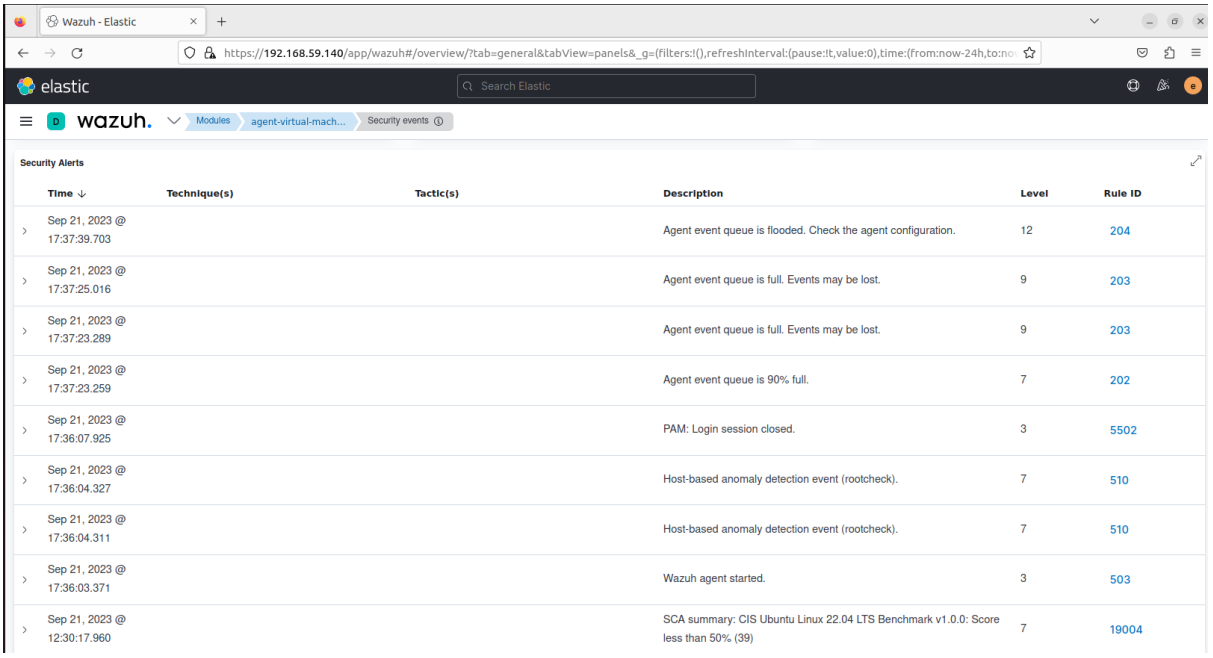
Also ensure Suricata is running and monitoring traffic on the correct network interface.

Test Wazuh Suricata Log Processing

It is now time to test if Wazuh can actually read and process Suricata event logs.

If Suricata is running on a live system with realtime traffic, there are high chances that you will sport the events related to network traffic on the respective Suricata server agent events.

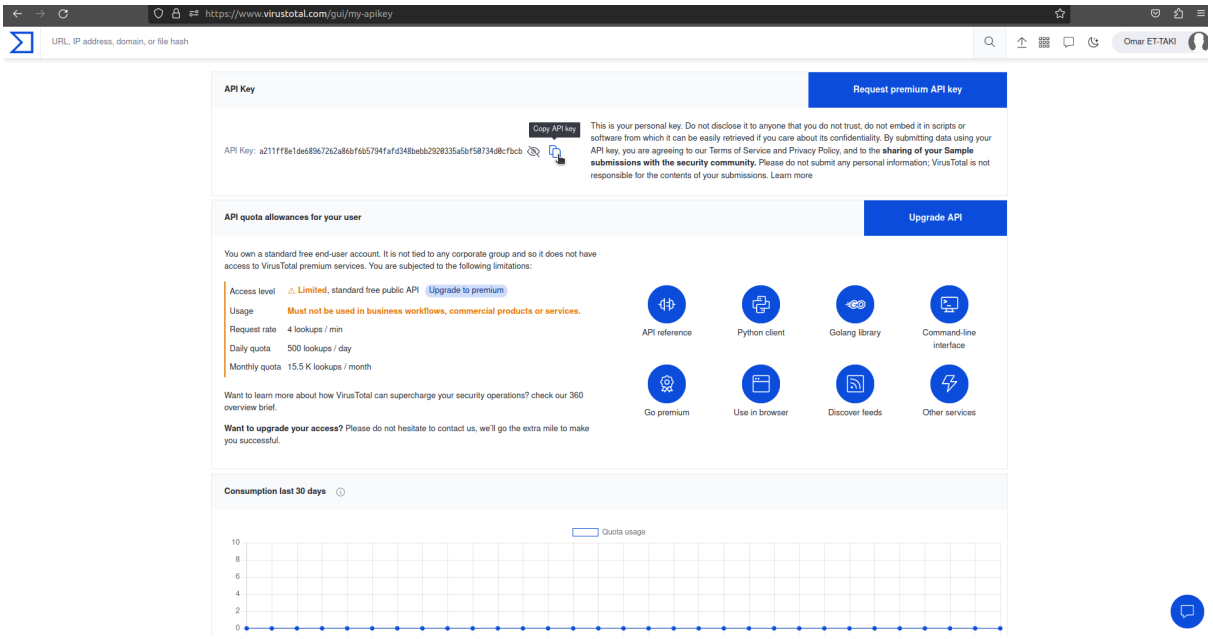
See sample screenshot below;



Integrate Wazuh Manager with VirusTotal

Obtain VirusTotal API Key

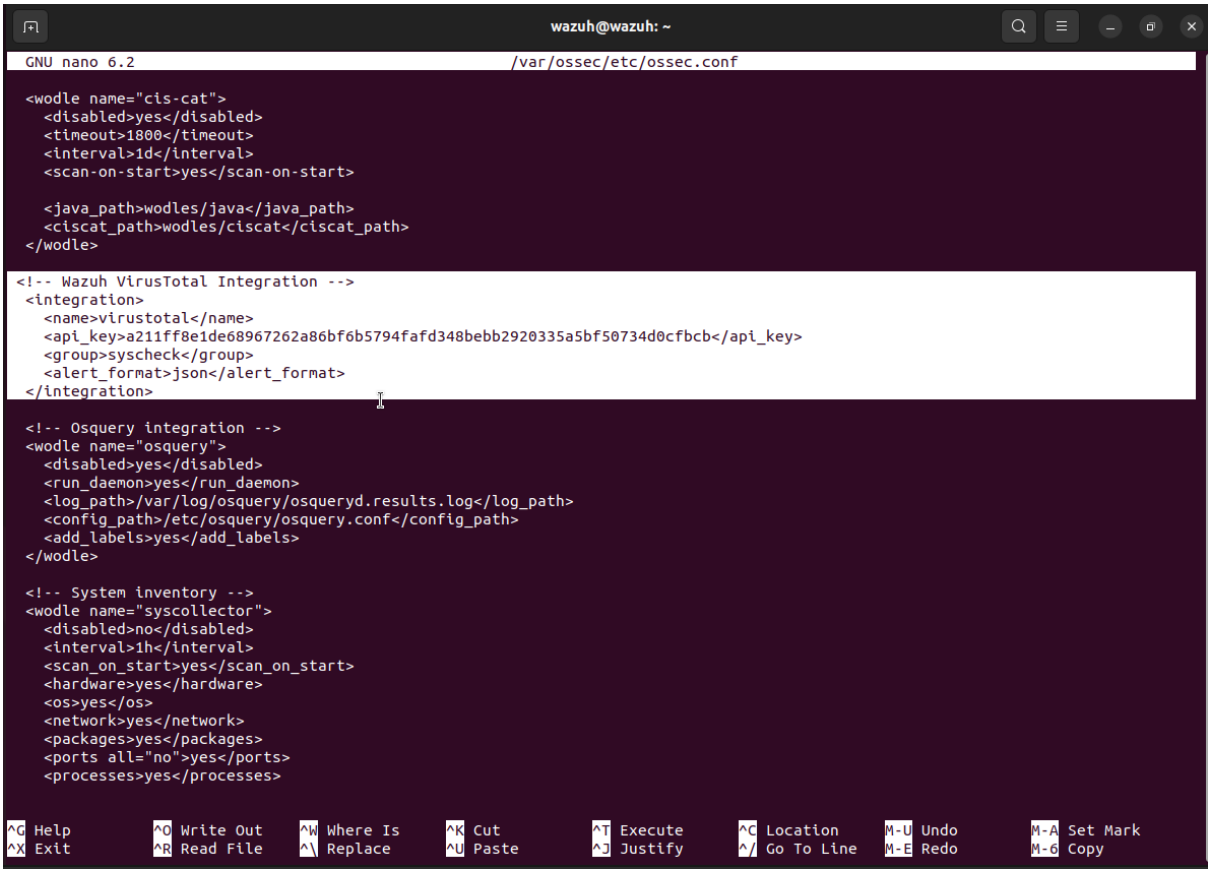
We need to obtain a VirusTotal API key. API key can be a public or a private one. We use a public one in this setup.
Once you have created an account and logged into VirusTotal, click on your username at the top right corner > API Key.



Integrate Wazuh Manager with VirusTotal

To integrate Wazuh manager with VirusTotal, you need to add the configuration below to /var/ossec/etc/ossec.conf file.

Replace the API_KEY with your respective VirusTotal API Key. For example, this is how my configuration is like.

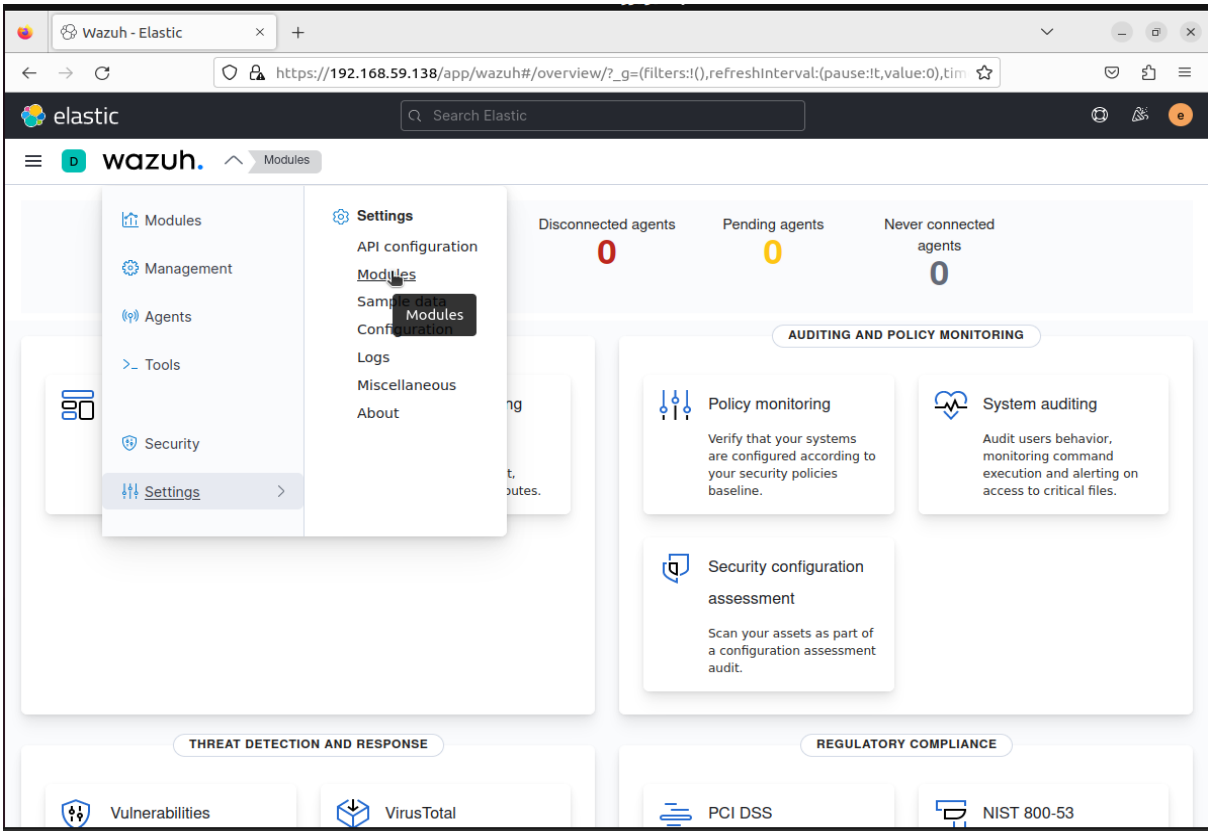


Once you have updated the configuration file, restart Wazuh manager service;

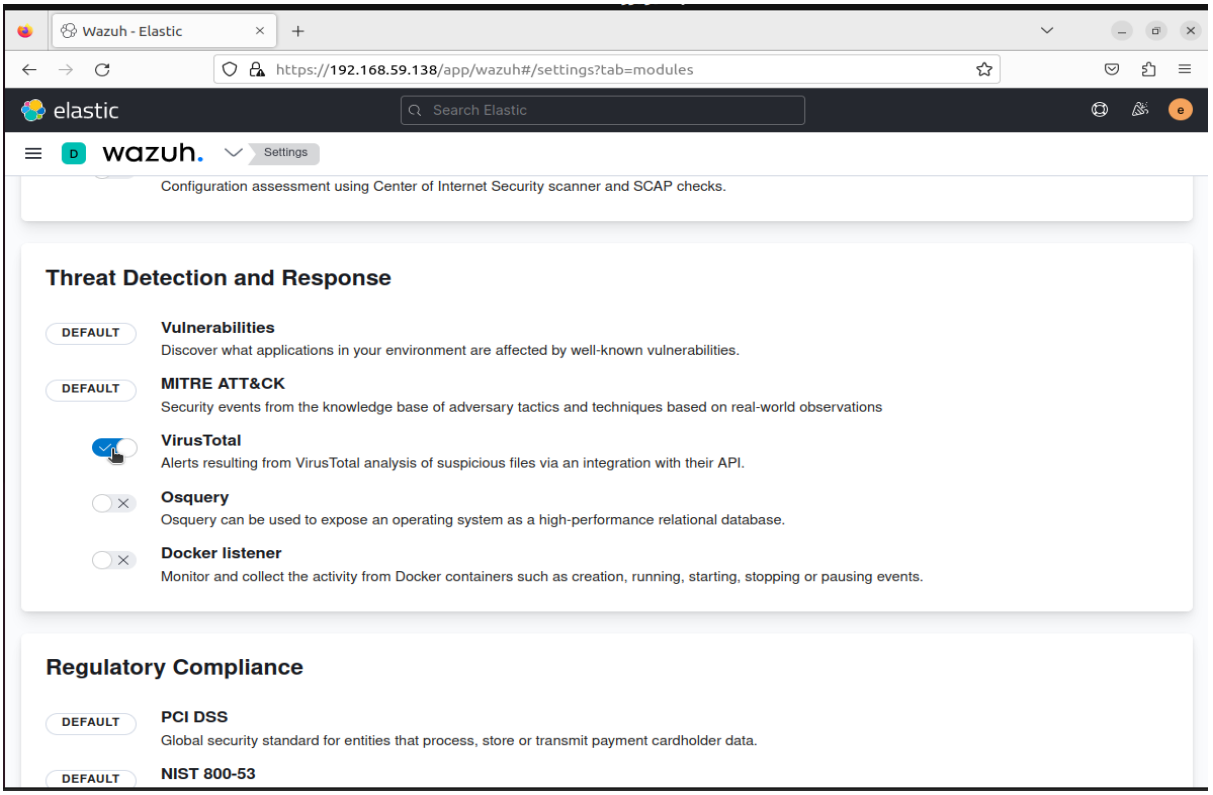
```
wazuh@wazuh:~$ sudo systemctl restart wazuh-manager
```

Enable Wazuh VirusTotal Module

Wazuh VirusTotal module is usually disabled by default. To enable the module, navigate to Kibana Web interface > Wazuh App > Wazuh Settings > Modules.



Scroll down to Threat Detection and Response section and toggle VirusTotal button to enable it.



Once you have enabled, you should now be able to access VirusTotal dashboard under Wazuh > Modules > Threat Detection and Response > VirusTotal.

For now, since we don’t already have any events, the dashboard is empty.

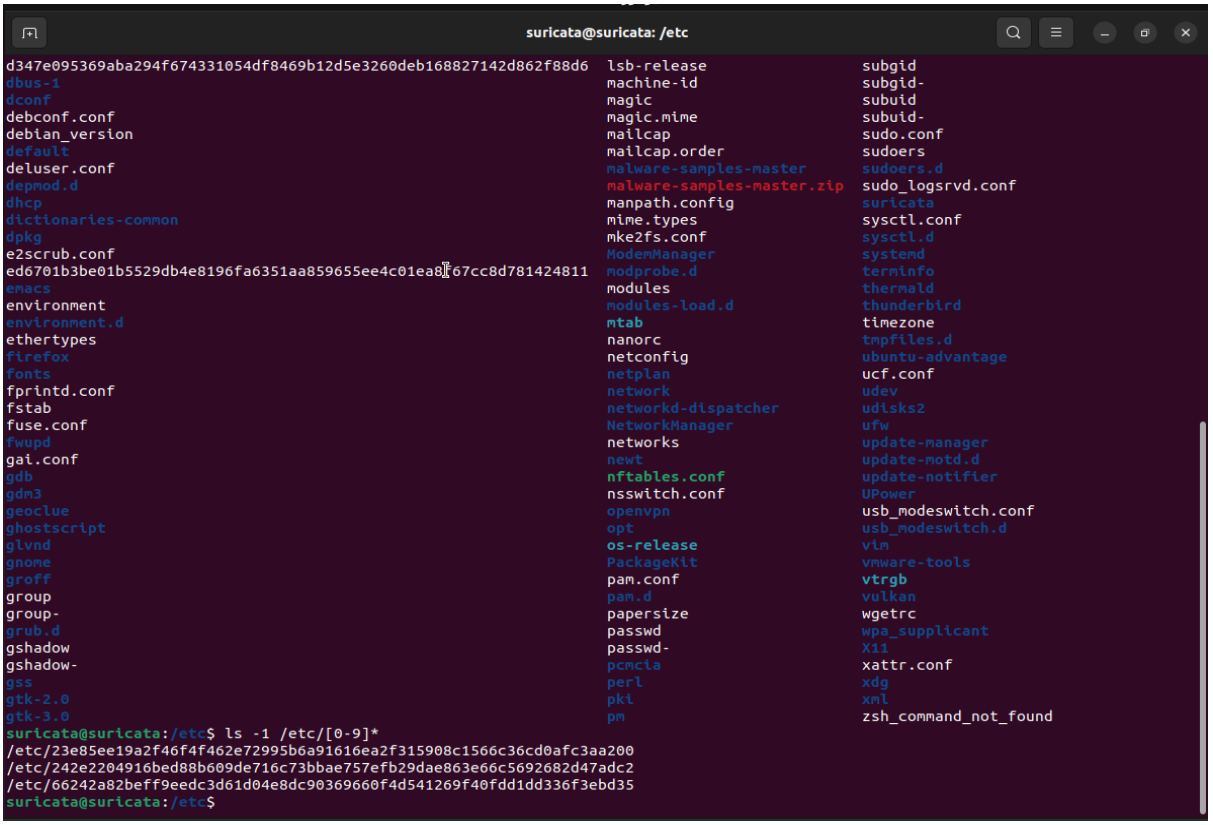
Demonstrating how Wazuh monitors the directories for Malicious Files

By default, Wazuh agent monitors a number of directories as defined on the Agents ossec.conf file as defined under the <!-- File integrity monitoring --> section.

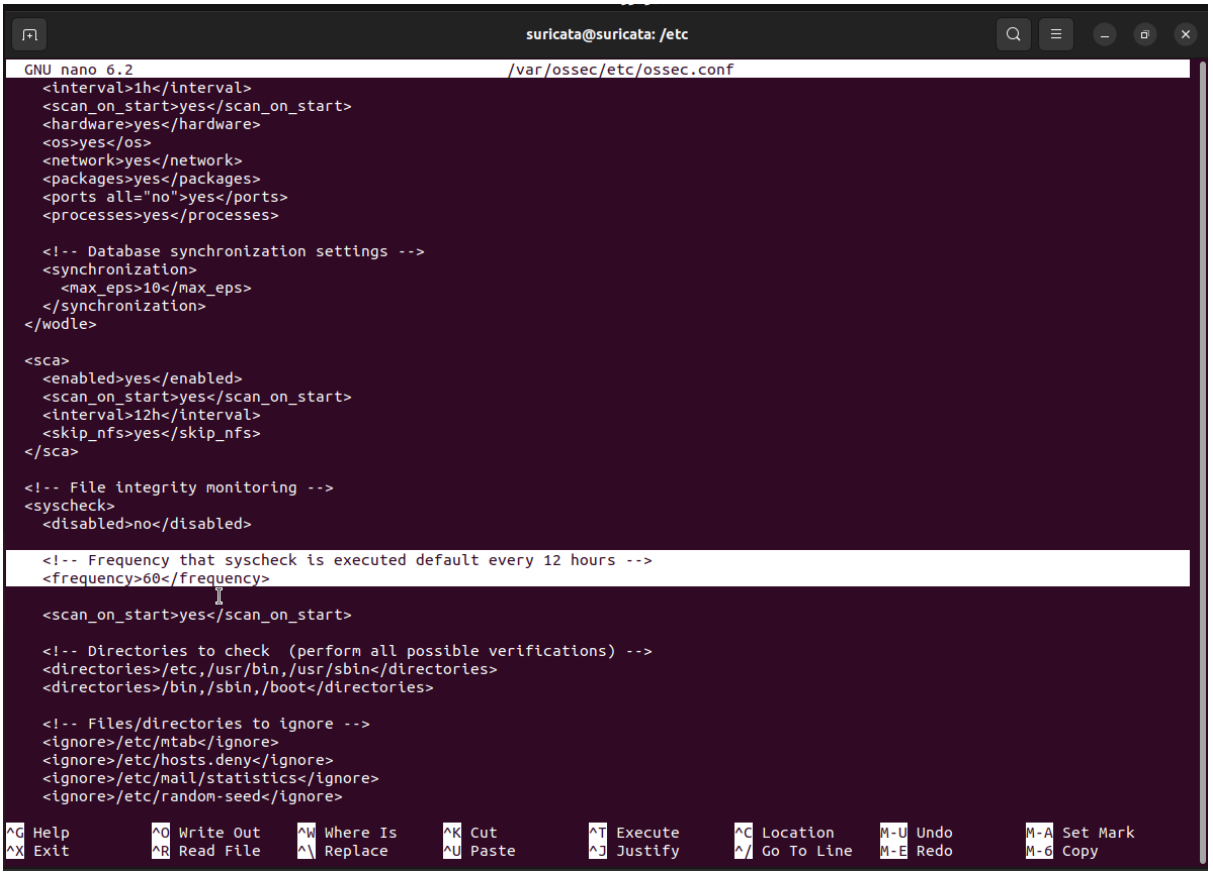
To effectively demonstrate this, we are gonna place a malicious file under the /etc directory and see what kind of events we get on the Wazuh dashboard.

NOTE: This is done on an isolated temporary test system. Do not place malicious files on your systems!!!! We cannot be held responsible on what malicious files may do to your system.

So as already mentioned that we are using a test system, we have downloaded some sample malware files from InQuest/malware-samples github repository to /etc directory of the Wazuh agent;



Note that we have also adjusted the Wazuh agent syscheck scan frequency from 12 hours to a min, just for demo purposes.

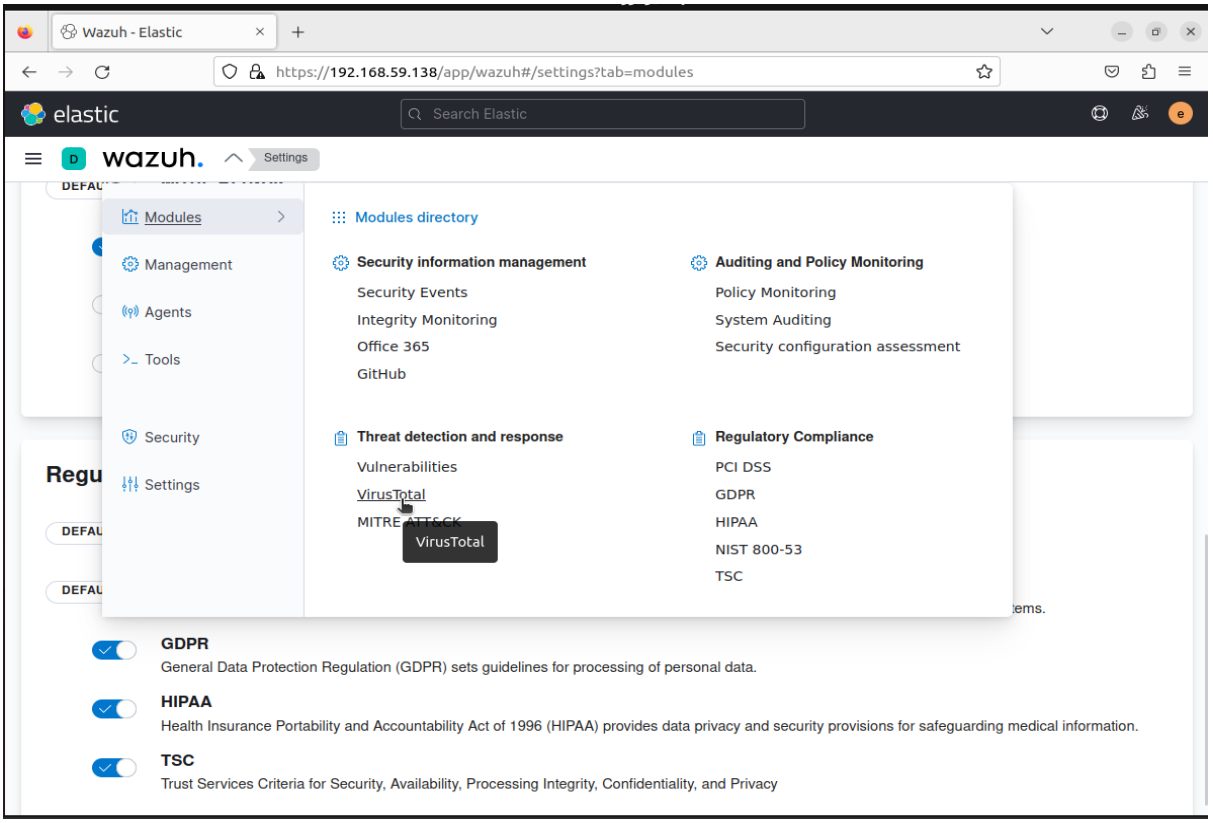


For any changes made to ossec.conf, you need to restart the agent.

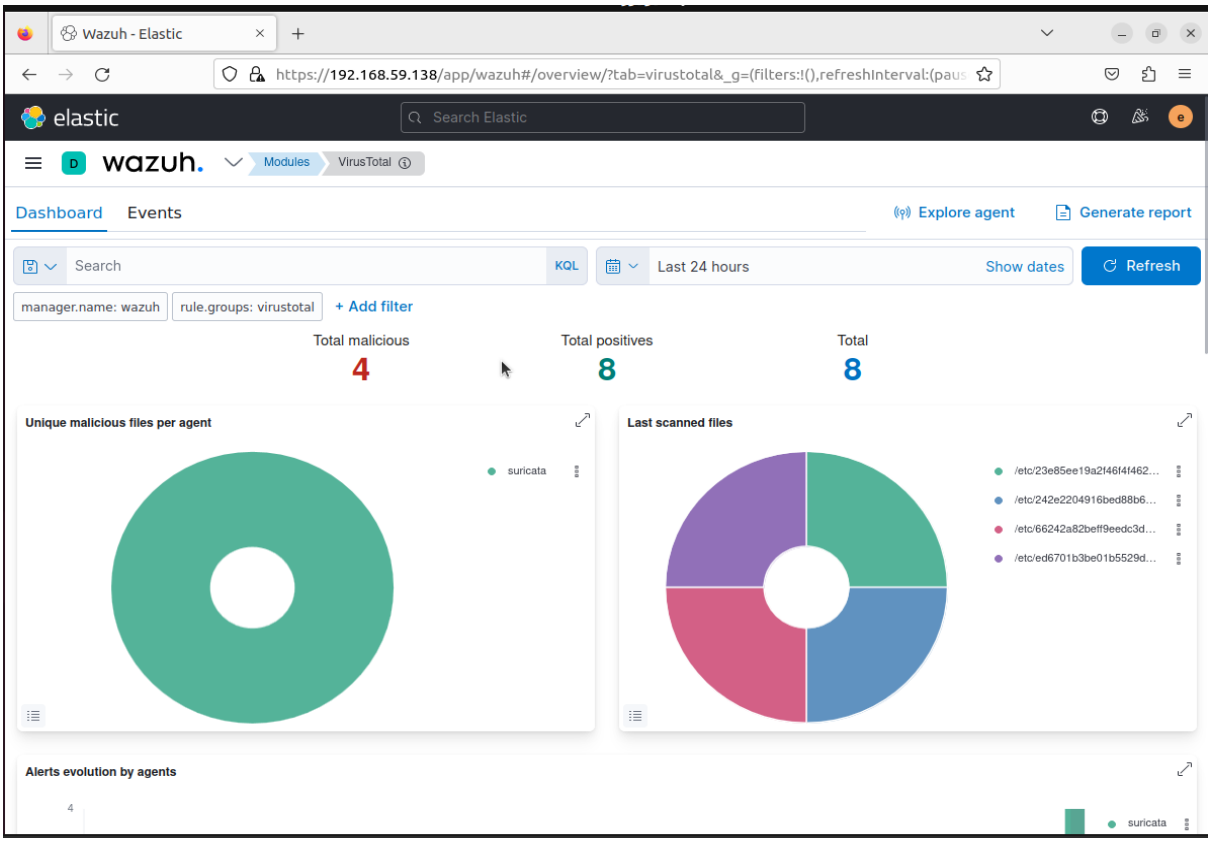
Verify Malware Detection Events on Wazuh Manager

Once the file is placed on the Wazuh agent system, you should be able to get events on the Wazuh manager dashboard.

Hence navigate Wazuh > Modules > Threat Detection and Response > VirusTotal.

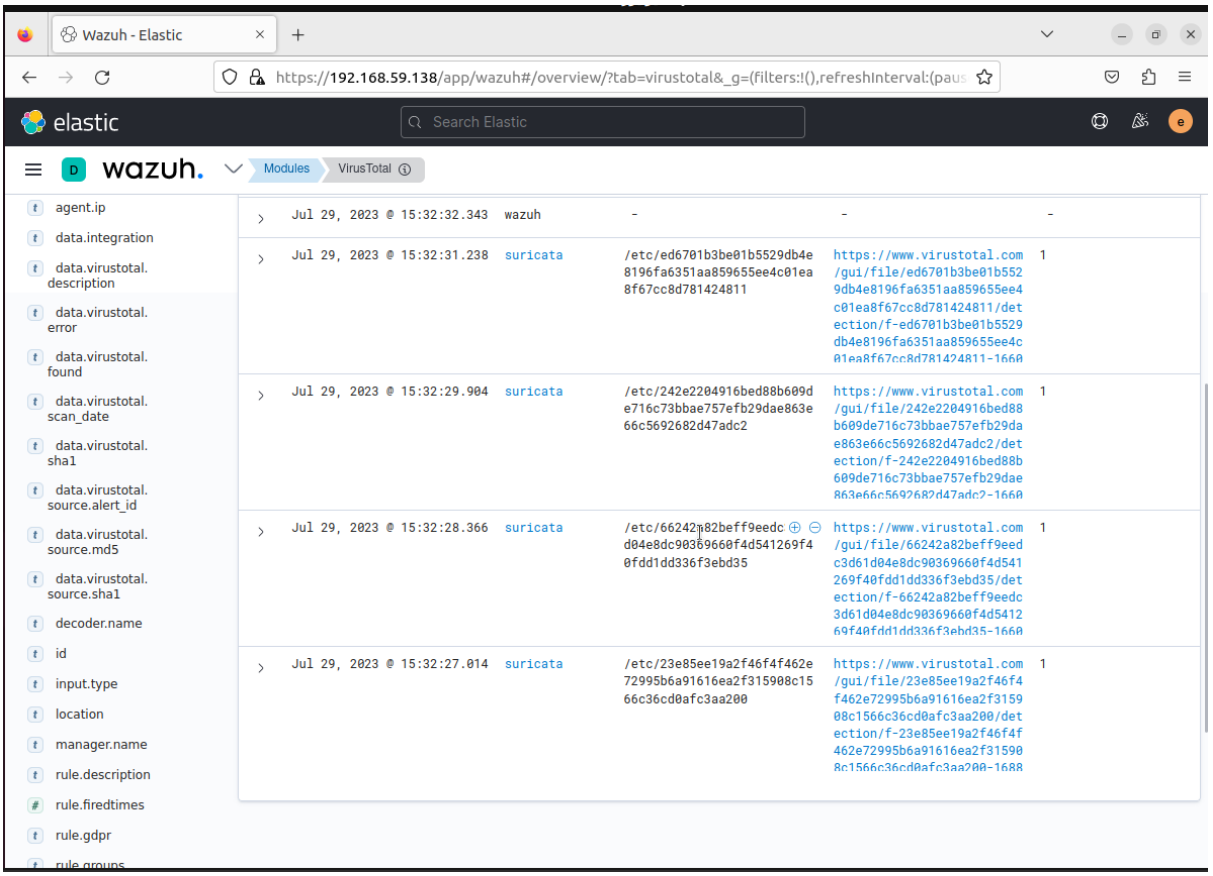


You can click Events to see the related events.



You can click the VirusTotal links to learn more about the files.

Also, if you check on Security Events, you should be able to see alerts related to files added to the system.



Nmap security auditing with Wazuh

Nmap integration

In this section, we run an Nmap scan using Python to provide information about open ports on an Ubuntu endpoint.

Nmap script

We created a Python script to perform network scans on an endpoint. The script extracts information such as hostnames, protocols, and open ports.

```
############################################################################

#!/var/ossec/framework/python/bin/python3

# Copyright (C) 2015-2023, Wazuh Inc.


import nmap

import time

import json

import platform


# The function to perform network scan on a host endpoint

def scan_subnet(subnet):

    nm = nmap.PortScanner()

    nm.scan(subnet)

    results = []


    for host in nm.all_hosts():

        for proto in nm[host].all_protocols():

            if proto not in ["tcp", "udp"]:
```

```
            continue


        lport = list(nm[host][proto].keys())
        lport.sort()
# Iterate over each port for the current host and protocol
        for port in lport:
            hostname = ""
            json_output = {
                'nmap_host': host,
                'nmap_protocol': proto,
                'nmap_port': port,
                'nmap_hostname': "",
                'nmap_hostname_type': "",
                'nmap_port_name': "",
                'nmap_port_state': "",
                'nmap_port_service': ""
            }
# Get the first hostname and it’s type
            if nm[host]["hostnames"]:
                hostname = nm[host]["hostnames"][0]["name"]
                hostname_type = nm[host]["hostnames"][0]["type"]
                json_output['nmap_hostname'] = hostname
                json_output['nmap_hostname_type'] = hostname_type
# Get the port name if available
            if 'name' in nm[host][proto][port]:
                json_output['nmap_port_name'] = nm[host][proto][port]['name']
```

```
# Get the port state if available
                if 'state' in nm[host][proto][port]:
                    json_output['nmap_port_state'] = nm[host][proto][port]['state']
# Get the port service version if available
                if 'product' in nm[host][proto][port] and 'version' in nm[host][proto][port]:
                    service = nm[host][proto][port]['product'] + " " + nm[host][proto][port]['version']
                    json_output['nmap_port_service'] = service


                results.append(json_output)
    return results


# The function to append the scan results to the active response log file
def append_to_log(results, log_file):
    with open(log_file, "a") as active_response_log:
        for result in results:
            active_response_log.write(json.dumps(result))
            active_response_log.write("\n")
# Specify the address(es) to scan
subnets = ['127.0.0.1']
# path of the log file
if platform.system() == 'Windows':
    log_file = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-responses.log"
elif platform.system() == 'Linux':
    log_file = "/var/ossec/logs/active-responses.log"
else:
    log_file = "/Library/Ossec/logs/active-responses.log"
```

```
for subnet in subnets:
    results = scan_subnet(subnet)
    append_to_log(results, log_file)
    time.sleep(2)
```

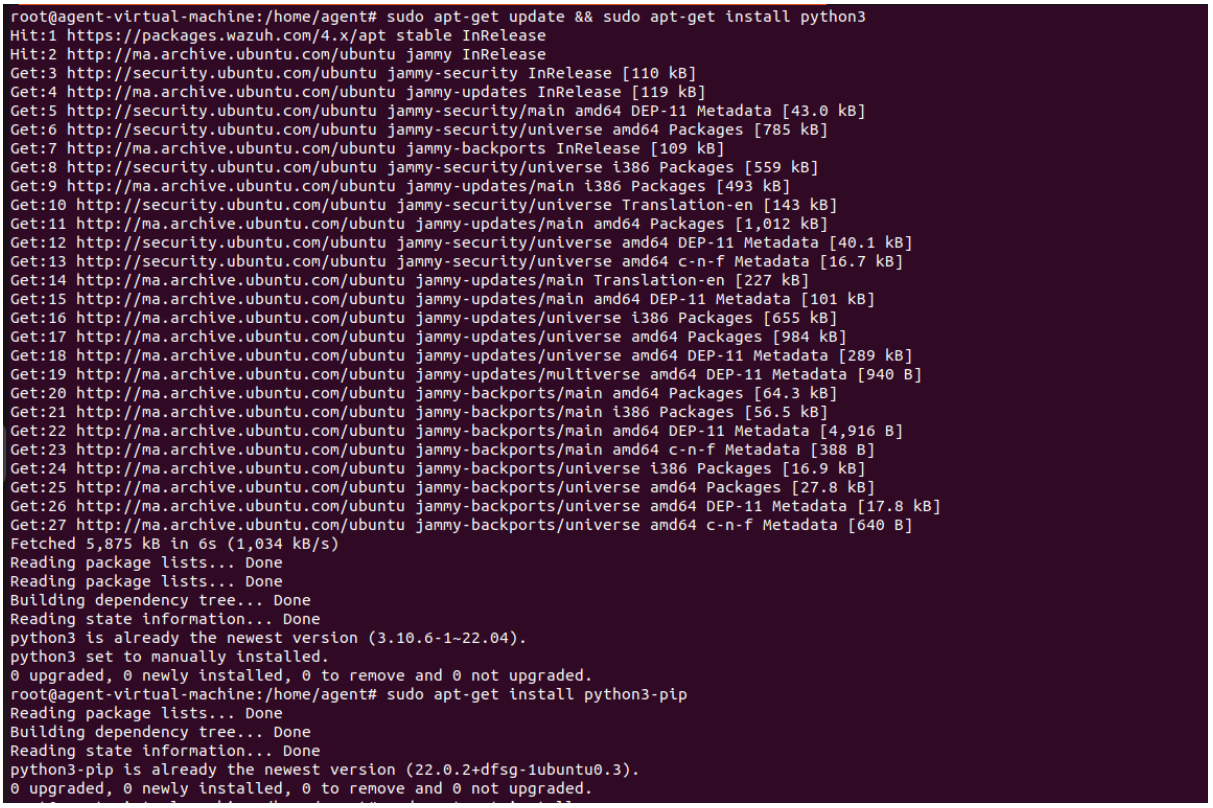
##

Ubuntu endpoint (wazuh agent):

Install the following packages to run an Nmap scan using Python on Ubuntu.

1. Install `python3` and `python3-pip` from the APT repository by running the command below:

sudo apt-get update && sudo apt-get install python3
sudo apt-get install python3-pip



2. Install `Nmap` and the `python-nmap` library. The `python-nmap` library provides many options for customizing Nmap scans:

sudo apt-get install nmap

```
root@agent-virtual-machine:/home/agent# sudo apt-get install nmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libblas3 liblinear4 lua-lpeg nmap-common
Suggested packages:
  liblinear-tools liblinear-dev ncat ndiff zenmap
The following NEW packages will be installed:
  libblas3 liblinear4 lua-lpeg nmap nmap-common
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,973 kB of archives.
After this operation, 26.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libblas3 amd64 3.10.0-2ubuntu1 [228 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy/universe amd64 liblinear4 amd64 2.3.0+dfsg-5 [41.4 kB]
Get:3 http://ma.archive.ubuntu.com/ubuntu jammy/universe amd64 lua-lpeg amd64 1.0.2-1 [31.4 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap-common all 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1 [3,940 kB]
Get:5 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap amd64 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1 [1,731 kB]
Fetched 5,973 kB in 2s (2,716 kB/s)
Selecting previously unselected package libblas3:amd64.
(Reading database ... 207019 files and directories currently installed.)
Preparing to unpack .../libblas3_3.10.0-2ubuntu1_amd64.deb ...
Unpacking libblas3:amd64 (3.10.0-2ubuntu1) ...
Selecting previously unselected package liblinear4:amd64.
Preparing to unpack .../liblinear4_2.3.0+dfsg-5_amd64.deb ...
Unpacking liblinear4:amd64 (2.3.0+dfsg-5) ...
Selecting previously unselected package lua-lpeg:amd64.
Preparing to unpack .../lua-lpeg_1.0.2-1_amd64.deb ...
Unpacking lua-lpeg:amd64 (1.0.2-1) ...
Selecting previously unselected package nmap-common.
Preparing to unpack .../nmap-common_7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1_all.deb ...
Unpacking nmap-common (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Selecting previously unselected package nmap.
Preparing to unpack .../nmap_7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1_amd64.deb ...
Unpacking nmap (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Setting up lua-lpeg:amd64 (1.0.2-1) ...
Setting up libblas3:amd64 (3.10.0-2ubuntu1) ...
update-alternatives: using /usr/lib/x86_64-linux-gnu/blas/libblas.so.3 to provide /usr/lib/x86_64-linux-gnu/libblas.so.3 (libblas.so.3-x86_64-linux-gnu) in auto mode
Setting up nmap-common (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Setting up liblinear4:amd64 (2.3.0+dfsg-5) ...
```

sudo pip3 install python-nmap

```
root@agent-virtual-machine:/home/agent# sudo pip3 install python-nmap
Collecting python-nmap
  Downloading python-nmap-0.7.1.tar.gz (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.4/44.4 KB 1.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: python-nmap
  Building wheel for python-nmap (setup.py) ... done
  Created wheel for python-nmap: filename=python_nmap-0.7.1-py2.py3-none-any.whl size=20637 sha256=5c6f4944684f68e42cfa8bf0553f911280f265e0394c427ee1a092584e152a88
  Stored in directory: /root/.cache/pip/wheels/da/bd/c6/0342ac886d4deb8d166a3191eb2566f738c5b1574cb0a8cd62
Successfully built python-nmap
Installing collected packages: python-nmap
Successfully installed python-nmap-0.7.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment
 instead: https://pip.pypa.io/warnings/venv
```

Take the following steps to configure the Wazuh command monitoring module.

1. Create a `Documents/nmapscan.py` file and copy the content of Nmap script to it.

```
  GNU nano 6.2                                                Documents/nmapscan.py
#!/var/ossec/framework/python/bin/python3
# Copyright (C) 2015-2023, Wazuh Inc.

import nmap
import time
import json
import platform

# The function to perform network scan on a host endpoint
def scan_subnet(subnet):
    nm = nmap.PortScanner()
    nm.scan(subnet)
    results = []

    for host in nm.all_hosts():
        for proto in nm[host].all_protocols():
            if proto not in ["tcp", "udp"]:
                continue

            lport = list(nm[host][proto].keys())
            lport.sort()
# Iterate over each port for the current host and protocol
            for port in lport:
                hostname = ""
                json_output = {
                    'nmap_host': host,
                    'nmap_protocol': proto,
                    'nmap_port': port,
                    'nmap_hostname': "",
                    'nmap_hostname_type': "",
                    'nmap_port_name': "",
                    'nmap_port_state': "",
                    'nmap_port_service': ""
                }
# Get the first hostname and it's type
                if nm[host]["hostnames"]:
                    hostname = nm[host]["hostnames"][0]["name"]
                    hostname_type = nm[host]["hostnames"][0]["type"]
                    json_output['nmap_hostname'] = hostname
                                                [ Read 74 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo      M-A Set Mark    M-] To Bracket  M-Q Previous
^X Exit      ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo      M-6 Copy        ^Q Where Was    M-W Next
```

2. Edit the Wazuh agent `/var/ossec/etc/ossec.conf` file and add the following command monitoring configuration within the `<ossec_config>` block:

##

<!-- Run nmap python script -->

<localfile>

<log_format>full_command</log_format>

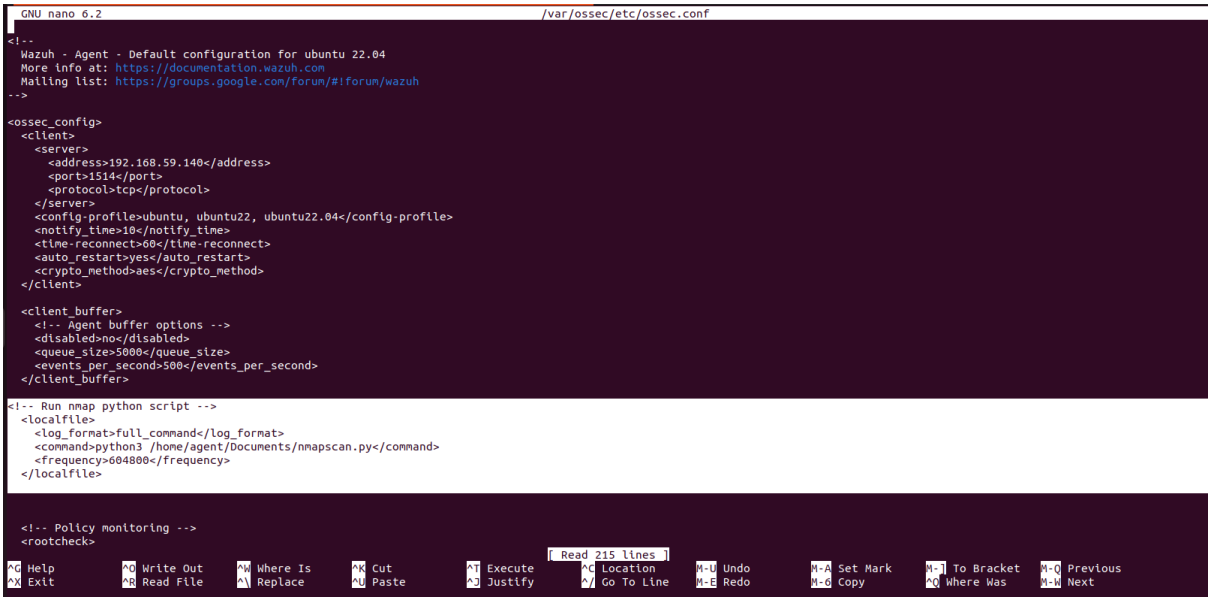
<command>python3 /home/<USERNAME>/Documents/nmapscan.py</command>

<frequency>604800</frequency>

</localfile>

##

Replace `<USERNAME>` placeholder with the name of the user account on the endpoint.



3. Restart the Wazuh agent to apply this change:

sudo systemctl restart wazuh-agent

Wazuh server :

In this section, we create a rule to capture the result of the Nmap scan on the monitored endpoint.

1. Add the rule below to the `/var/ossec/etc/rules/local_rules.xml` file:

##

<group name="linux,nmap,">

<rule id="100100" level="3">

<decoded_as>json</decoded_as>

<field name="nmap_port">\.+</field>

<field name="nmap_port_service">\.+</field>

<description>NMAP: Host scan. Port $(nmap_port) is open and hosting the $(nmap_port_service) service.</description>

<options>no_full_log</options>

</rule>

</group>

##

Where:

- Rule ID `100100` is triggered after a successful Nmap scan on the monitored endpoint.

2. Restart the Wazuh manager to apply the configuration changes:

sudo systemctl restart wazuh-manager

Scan results

The images below show the alerts generated on the Wazuh dashboard when we perform Nmap scans on the Ubuntu endpoint.

Navigate to the **Security events** tab to view the generated alerts.