

Report:

WAZUH

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

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
```

```
root@wazuh:/home/wazuh# apt-get install apt-transport-https zip unzip lsb-release curl gnupg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release set to manually installed.
zip is already the newest version (3.0-12build2).
zip set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.13).
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
unzip is already the newest version (6.0-26ubuntu3.1).
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,510 B of archives.
After this operation, 169 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.9 [1,510 B]
Fetched 1,510 B in 0s (3,235 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 197974 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.9_all.deb ...
Unpacking apt-transport-https (2.4.9) ...
Setting up apt-transport-https (2.4.9) ...
```

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 D27D0666C088E42B4: 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/elastic-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 1386 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
```

```
root@wazuh:/home/wazuh# /usr/share/elasticsearch/bin/elasticsearch-certutil cert ca --pem --in instances.yml --keep-ca-key --out ~/certs.zip  
This tool assists you in the generation of X.509 certificates and certificate  
signing requests for use with SSL/TLS in the Elastic Stack.  
The 'cert' mode generates X.509 certificate and private keys.  
* By default, this generates a single certificate and key for use  
on a single instance.  
* The '-multiple' option will prompt you to enter details for multiple  
instances and will generate a certificate and key for each one.  
* The '--in' option allows for the certificate generation to be automated by describing  
the details of each instance in a YAML file.  
* An instance is any piece of the Elastic Stack that requires an SSL certificate.  
Depending on your configuration, Elasticsearch, Logstash, Kibana, and Beats  
may all require a certificate and private key.  
* The minimum required value for each instance is a name. This can simply be the  
hostname, which will be used as the Common Name of the certificate. A full  
distinguished name may also be used.  
* A filename value may be required for each instance. This is necessary when the  
name would result in an invalid file or directory name. The name provided here  
is used as the directory name (within the zip) and the prefix for the key and  
certificate files. The filename is required if you are prompted and the name  
is not displayed in the prompt.  
* IP addresses and DNS names are optional. Multiple values can be specified as a  
comma separated string. If no IP addresses or DNS names are provided, you may  
disable hostname verification in your SSL configuration.  
* All certificates generated by this tool will be signed by a certificate authority (CA)  
unless the --self-signed command line option is specified.  
The tool can automatically generate a new CA for you, or you can provide your own with  
the --ca or --ca-cert command line options.  
By default the 'cert' mode produces a single PKCS#12 output file which holds:  
* The instance certificate  
* The private key for the instance certificate  
* The CA certificate  
If you specify any of the following options:  
* -pem (PEM formatted output)  
* -keep-ca-key (retain generated CA key)  
* -multiple (generate multiple certificates)  
* -in (generate certificates from an input file)  
then the output will be a zip file containing individual certificate/key files  
Note: Generating certificates without providing a CA certificate is deprecated.  
A CA certificate will become mandatory in the next major release.
```

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

```
# unzip ~/certs.zip -d ~/certs
```

```
root@wazuh:/home/wazuh# unzip ~/certs.zip -d ~/certs  
Archive: /root/certs.zip  
  creating: /root/certs/ca/  
  inflating: /root/certs/ca/ca.crt  
  inflating: /root/certs/ca/ca.key  
  creating: /root/certs/elasticsearch/  
  inflating: /root/certs/elasticsearch/elasticsearch.crt  
  inflating: /root/certs/elasticsearch/elasticsearch.key
```

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 elastic:elastic /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 -Xssim -Dj...  
-4363 /usr/share/elasticsearch/modules/x-pack/ml/platform/linux-x86_64/bin/controller

11:55:10 20 j... wazuh systemd[1]: Starting Elasticsearch...
11:55:25 20 j... 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]

Changed password for user apm_system
PASSWORD apm_system = 4qxO20lpmHYpFnczBCv

Changed password for user kibana_system
PASSWORD kibana_system = WU8ly9jN88gNHqDXULE

Changed password for user kibana
PASSWORD kibana = WU8ly9jN88gNHqDXULE

Changed password for user logstash_system
PASSWORD logstash_system = kanPwbsz38UuwuDKwMfg

Changed password for user beats_system
PASSWORD beats_system = jDCvnX0hk8TQ3xp8CSI4

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" : "ef4822227ee6b9e70e502f0f0daa52435ee634d",
    "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 9683EEF29111145: 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
                  └─48995 /var/ossec/bin/wazuh-wazuhd

Jul 20 12:07:55 20 يو[48707] يو[48707]: Started wazuh-db...
Jul 20 12:07:56 20 يو[48707] يو[48707]: Started wazuh-execd...
Jul 20 12:07:58 20 يو[48707] يو[48707]: Started wazuh-analysisd...
Jul 20 12:07:59 20 يو[48707] يو[48707]: Started wazuh-syscheckd...
Jul 20 12:07:59 20 يو[48707] يو[48707]: Started wazuh-remoted...
Jul 20 12:07:59 20 يو[48707] يو[48707]: Started wazuh-logcollector...
Jul 20 12:08:00 20 يو[48707] يو[48707]: Started wazuh-monitord...
Jul 20 12:08:00 20 يو[48707] يو[48707]: Started wazuh-modulesd...
Jul 20 12:08:02 20 يو[48707] يو[48707]: Completed.
Jul 20 12:08:02 20 يو[48707] يو[48707]: 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
root@wazuh:/home/wazuh#
```

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
```

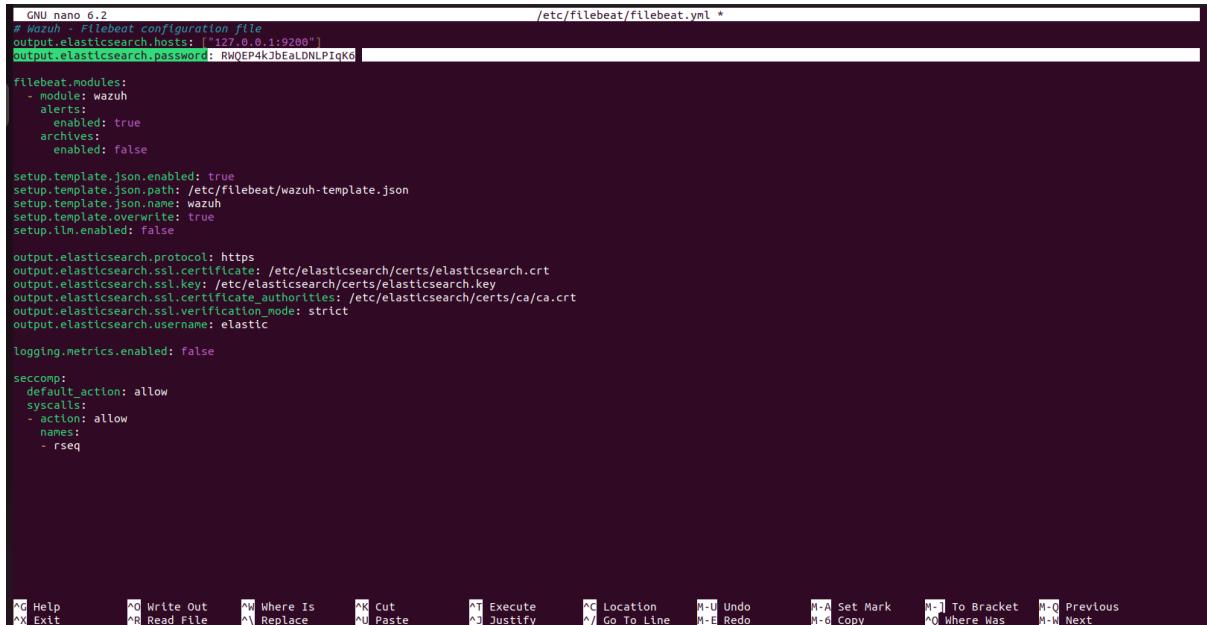
4. Download the Wazuh module for Filebeat:

```
# 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>
```



```
GNU nano 6.2                               /etc/filebeat/filebeat.yml *
# Wazuh - Filebeat configuration file
output.elasticsearch.hosts: ["127.0.0.1:9200"]
output.elasticsearch.password: RWQEP4kJbEaLDNLPIqK6

filebeat.modules:
- module: wazuh
  alerts:
    enabled: true
  archives:
    enabled: false

setup.template.json.enabled: true
setup.template.json.path: /etc/filebeat/wazuh-template.json
setup.template.json.name: wazuh
setup.template.overwrite: true
setup.tlm.enabled: false

output.elasticsearch.protocol: https
output.elasticsearch.ssl.certificate: /etc/elasticsearch/certs/elasticsearch.crt
output.elasticsearch.ssl.key: /etc/elasticsearch/certs/elasticsearch.key
output.elasticsearch.ssl.certificateAuthorities: /etc/elasticsearch/certs/ca/ca.crt
output.elasticsearch.ssl.verification_mode: strict
output.elasticsearch.username: elastic

logging.metrics.enabled: false

seccomp:
  default_action: allow
  syscalls:
  - action: allow
    names:
    - rseq

^G Help      ^O Write Out     ^M Where Is      ^K Cut          ^I Execute      ^C Location      ^U Undo        ^T Set Mark     ^I To Bracket   ^Q Previous
^X Exit      ^R Read File     ^H Replace       ^V Paste         ^A Justify      ^L Go To Line    ^E Redo        ^D Copy        ^W Where Was    ^H Next
```

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
```

```
root@wazuh:/home/wazuh# cp -r /etc/elasticsearch/certs/ca/ /etc/filebeat/certs/
root@wazuh:/home/wazuh# cp /etc/elasticsearch/certs/elasticsearch.crt /etc/filebeat/certs/filebeat.crt
root@wazuh:/home/wazuh# 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
```

```
root@wazuh:/home/wazuh# systemctl daemon-reload
root@wazuh:/home/wazuh# systemctl enable filebeat
Synchronizing state of filebeat service with SysV service script with /lib/systemd/systemd-sysv-install.
Created symlink /lib/systemd/systemd-sysv-install enable filebeat.
root@wazuh:/home/wazuh# systemctl start filebeat
root@wazuh:/home/wazuh# systemctl status filebeat
● filebeat.service - filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-07-20 12:19:33 +01; 6s ago
       Docs: https://www.elastic.co/beats/filebeat
      Main PID: 51059 (filebeat)
        Tasks: 7 (limit: 4600)
       Memory: 37.1M
          CPU: 150ms
         CGroup: /system.slice/filebeat.service
                 └─51059 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/share/filebeat --path.config /etc/filebeat --path.data /var/lib/filebeat

12:19:34.20 wazuh filebeat[51059]: 2023-07-20T12:19:34.335+0100      INFO      [publisher]      pipeline/retry.go:219      retryer: send unpaid signal to consumer
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.335+0100      INFO      [publisher]      pipeline/retry.go:223      done
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.358+0100      INFO      [esclientleg]      eslegclient/connection.go:285      Attempting to connect to Ela...
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.430+0100      INFO      [esclientleg]      eslegclient/connection.go:285      Attempting to connect to Ela...
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.433+0100      INFO      template/load.go:197      Existing template will be overwritten, as overwrite is ena...
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.434+0100      INFO      template/load.go:131      Try loading template wazuh to Elasticsearch
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.606+0100      INFO      template/load.go:123      Template with name "wazuh" loaded.
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.607+0100      INFO      [index-management]      idxmgmt/std.go:297      Loaded index template.
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.664+0100      INFO      [modules]      fileset/pipelines.go:133      Elasticsearch pipeline loaded.
12:19:34.20 بولموز wazuh filebeat[51059]: 2023-07-20T12:19:34.664+0100      INFO      [publisher_pipeline_output]      pipeline/output.go:151      Connection to backoff...
lines 1-21 (END)
```

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

```
# filebeat test output
```

```
root@wazuh:/home/wazuh# filebeat test output
elasticsearch: https://127.0.0.1:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 127.0.0.1
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.17.9
```

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:

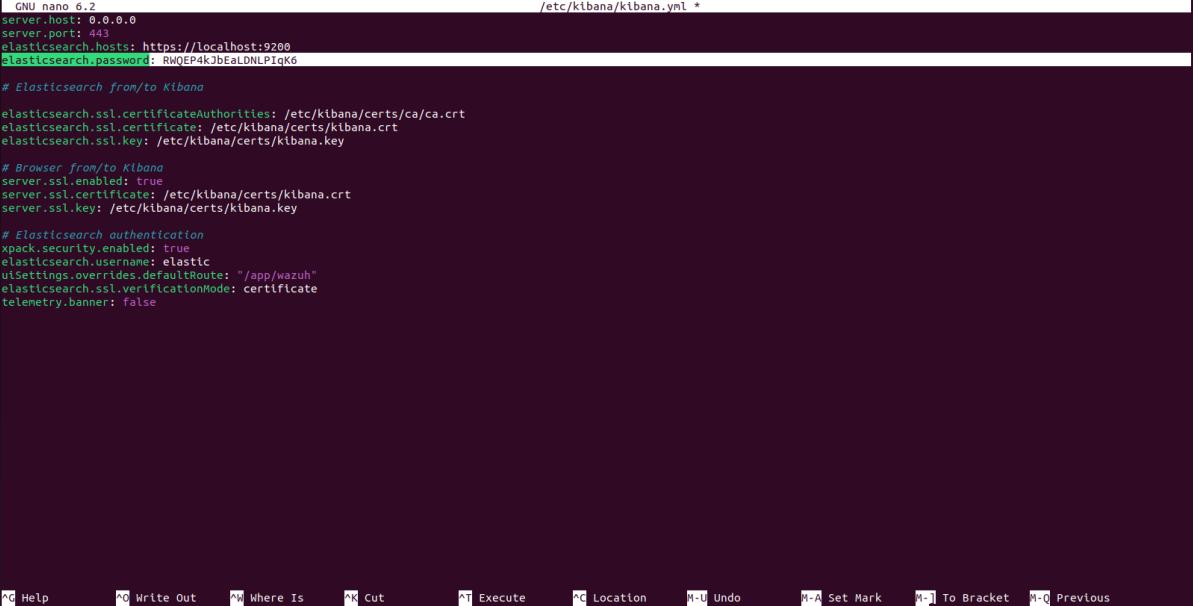
```
# curl -so /etc/kibana/kibana.yml
https://packages.wazuh.com/4.4/tpl/elastic-basic/kibana_all_in_one.yml
```

```
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.



```
GNU nano 6.2                               /etc/kibana/kibana.yml *

server.host: 0.0.0.0
server.port: 443
elasticsearch.hosts: https://localhost:9208
elasticsearch.password: RWQEP4kObEaLDNLPIqK6

# Elasticsearch from/to Kibana
elasticsearch.ssl.certificateAuthorities: /etc/kibana/certs/ca/ca.crt
elasticsearch.ssl.certificate: /etc/kibana/certs/kibana.crt
elasticsearch.ssl.key: /etc/kibana/certs/kibana.key

# Browser from/to Kibana
server.ssl.enabled: true
server.ssl.certificate: /etc/kibana/certs/kibana.crt
server.ssl.key: /etc/kibana/certs/kibana.key

# Elasticsearch authentication
xpack.security.enabled: true
elasticsearch.username: elastic
uiSettings.overrides.defaultRoute: "/app/wazuh"
elasticsearch.ssl.verificationMode: certificate
telemetry.banner: false
```

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 "-"

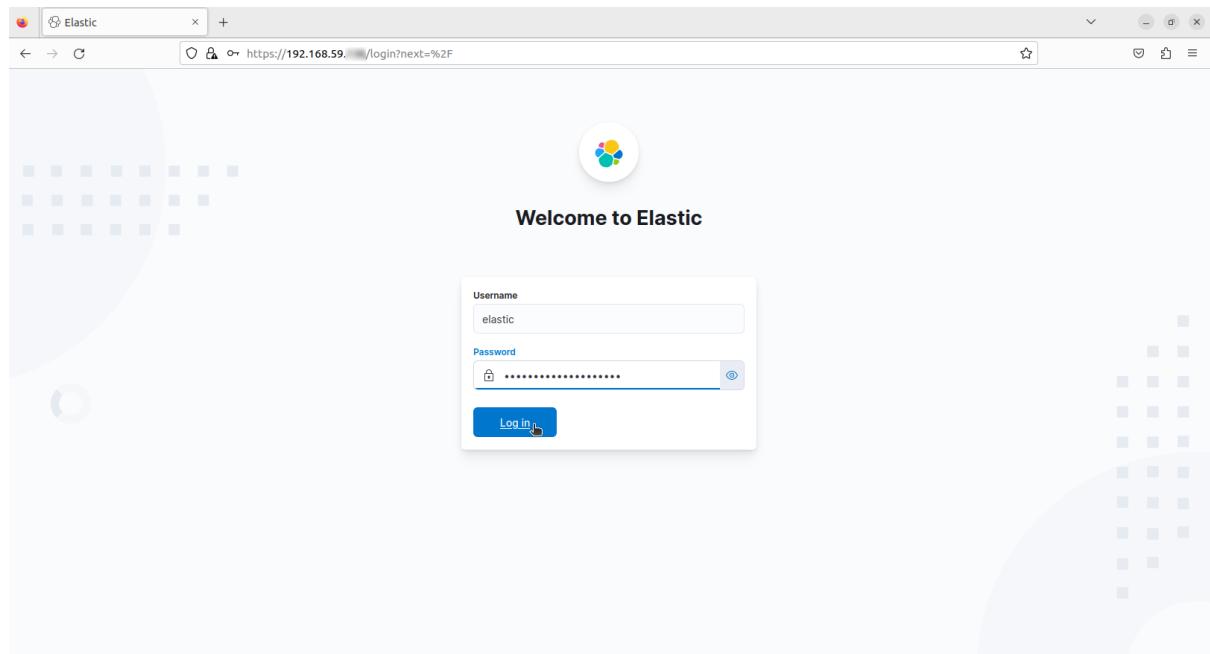
12:48:46.20 jsl_12 Wazuh systemd[1]: Started Kibana.
lines 1-12/12 (END)
```

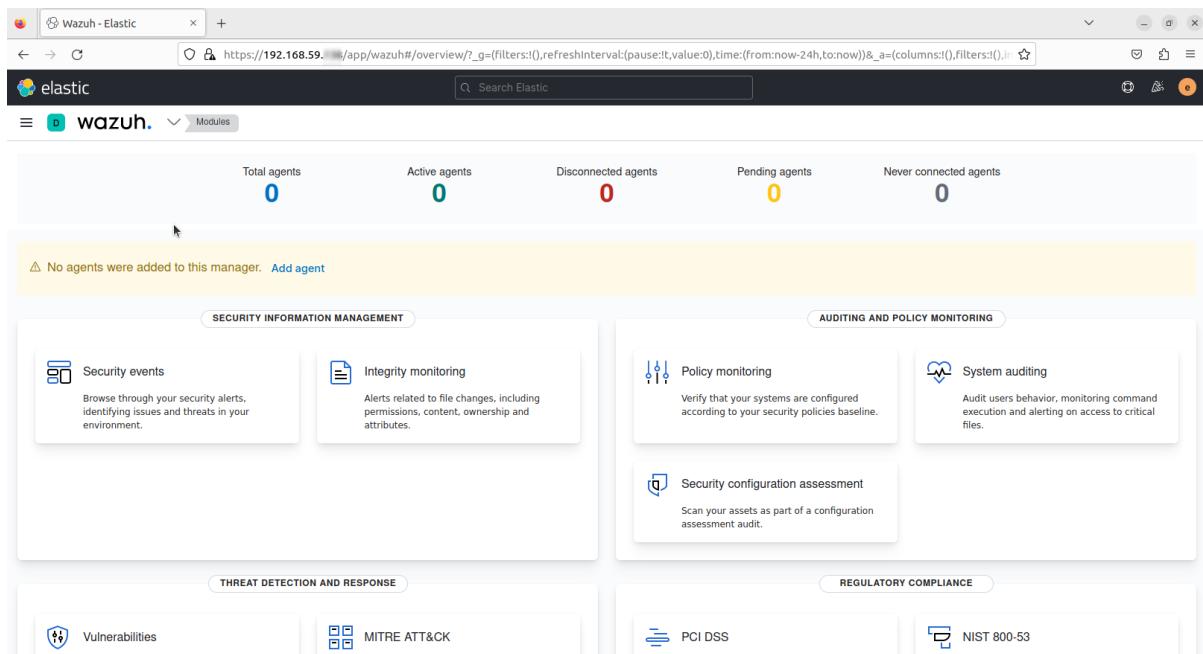
8. Access the web interface using the password generated during the Elasticsearch installation process:

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 96B3E5F29111145: 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 ... 198408 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
           ├─3770 /var/ossec/bin/wazuh-agend
           ├─3798 /var/ossec/bin/wazuh-syscheckd
           ├─3802 /var/ossec/bin/wazuh-logcollector
           └─3819 /var/ossec/bin/wazuh-modulesd
```

the Wazuh dashboard

The screenshot shows the Wazuh dashboard interface. At the top, there's a navigation bar with a logo, search bar, and various icons. Below it, the main dashboard area is divided into three main sections:

- STATUS:** A large teal circle icon with a white outline. To its right, a legend indicates:
 - Active (1)
 - Disconnected (0)
 - Pending (0)
 - Never connected (0)
- DETAILS:** Displays agent statistics:
 - Active: 1
 - Disconnected: 0
 - Pending: 0
 - Never connected: 0

Agents coverage: **100.00%**

Last registered agent: **agent-virtual-mach...**

Most active agent: **agent-virtual-machine**
- EVOLUTION:** A line chart titled "Last 24 hours" showing the count of events over time. The Y-axis is labeled "Count" and ranges from 0 to 1. The X-axis is labeled "timestamp per 10 minutes". A single yellow dot is plotted at approximately 0.8 on the count axis at the 10:00 mark.

At the bottom, there's a search/filter bar and a "Refresh" button. Below that, a table titled "Agents (1)" lists the single active agent:

ID	Name	IP address	Group(s)	Operating system	Cluster node	Version	Status	Actions	
001	agent-virtual-machine	192.168.59.141	default	Ubuntu 22.04.3 LTS	node01	v4.5.2	active		

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 integrate 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/agents# apt -y install libpcr3 libpcr3-dbg libpcr3-dev \
build-essential autoconf libtinfo libpcap-dev \
libneti-dev libyaml-0-2 libyaml-dev zlib zlibng-dev \
libpcap-ng-dev libcap-ng0 make libmagic-dev \
libjansson-dev libjansson4 pkg-config libnspir4-dev \
libnss3-dev libl24-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.iubuntud3).
libjansson4 set to manually installed.
libyaml0-0-2 is already the newest version (0.2.2-1ubuntud2).
libyaml0-0-2 set to manually installed.
libpcr3-1 is already the newest version (2:8.39-13ubuntu0.22.04.1).
libpcr3 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++-11 gcc gcc-11 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan0 libbinutils libdev-bin libdev-tools libbc1c0_0 libcrypt-dev liblbcftf0 liblbcftf0 liblbusbs-1-dev liblbpkg liblexpat1-dev
  libfakeroot liblfile-fcntllock-perl liblbc1c-11-dev liblbitm1 libljs-jquery liblsphnixdoc liblsunderscore liblsan0 libltdl-dev libneti1 liblinsn1-dev libpcap0_8-dev libpcr16-3
  libpcr3-2 libpcrcpp0v5 libpython3.10-dev libquadrath0 libsigsegv2 libssh2-1 libstdc-rust-1.66 libstdc-rust-dev libstdc++-11-dev libltlrcp-dev libtsan0
  libubsan1 linux-libc-dev libltdl-disabled-list m4 manpages-dev python3-dev 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 gcc-11-doc gcc-multilib flex bison gcc-doc
  g++-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
  python3-sphinx-doc tcl-dev lang-15
The following NEW packages will be installed:
  autotools-dev binutils binutils-common binutils-x86_64-linux-gnu dpkg-dev fakeroot g++-11 gcc gcc-11 javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libasan0 libbinutils libdev-bin libdev-tools libbc1c0_0 libcrypt-dev liblbcftf0 liblbcftf0 liblbusbs-1-dev liblbpkg liblexpat1-dev
  libfakeroot liblfile-fcntllock-perl liblbc1c-11-dev liblbitm1 libljs-jquery liblsphnixdoc liblsunderscore liblsan0 libltdl-dev libneti1 liblinsn1-dev libpcap0_8-dev libpcr16-3
  libpcr3-2 libpcrcpp0v5 libpython3.10-dev libquadrath0 libsigsegv2 libssh2-1 libstdc-rust-1.66 libstdc-rust-dev libstdc++-11-dev libltlrcp-dev
  libtsan0 libubsan1 linux-libc-dev libltdl-disabled-list m4 manpages-dev pkg-config python3-dev python3-distutils python3-pip python3-setup tools
  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 2.13-1ubuntud3 [14.6 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 m4 amd64 1.4.18-Subuntu2 [199 kB]
```

Suricata function as an IDS out of the box. If you need to include the IPS functionality, install the following libraries.

```
# sudo apt -y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink
```

```
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 (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_amd64_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) ...
```

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 ~]# 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, 2606:if16: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 libhttp
make[2]: Entering directory '/home/agent/suricata-6.0.5/libhttp'
Making install in http
make[3]: Entering directory '/home/agent/suricata-6.0.5/libhttp/http'
Making install in lzma
make[4]: Entering directory '/home/agent/suricata-6.0.5/libhttp/http/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-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 -MT LzFind.lo -MD -MP -MF .deps/LzFind.Tpo -o LzFind.lo 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-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 -o LzFind.lo 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-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 -o LzFind.o /dev/null 2>1
mv .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-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 -MT LznaDec.lo -MD -MP -MF .deps/LznaDec.Tpo -o LznaDec.lo LznaDec.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-arith -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC -MT LznaDec.lo -MD -MP -MF .deps/LznaDec.Tpo -o LznaDec.o /dev/null 2>1
mv .deps/LznaDec.Tpo .deps/LznaDec.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 LznaDec.lo -lz
libtool: link: ar cr liblzma-c.a .libs/LzFind.o .libs/LznaDec.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/libhttp/http/lzma'
make[5]: Nothing to be done for 'install-exec-am'.
/usr/bin/mkdir -p /usr/include/http/lzma
/usr/bin/install -c -D 644 LznaDec.h 7ztypes.h '/usr/include/http/lzma'
make[5]: Leaving directory '/home/agent/suricata-6.0.5/libhttp/http/lzma'
make[4]: Entering directory '/home/agent/suricata-6.0.5/libhttp/http'
/bin/bash ../../libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I . -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 -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 protected 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.

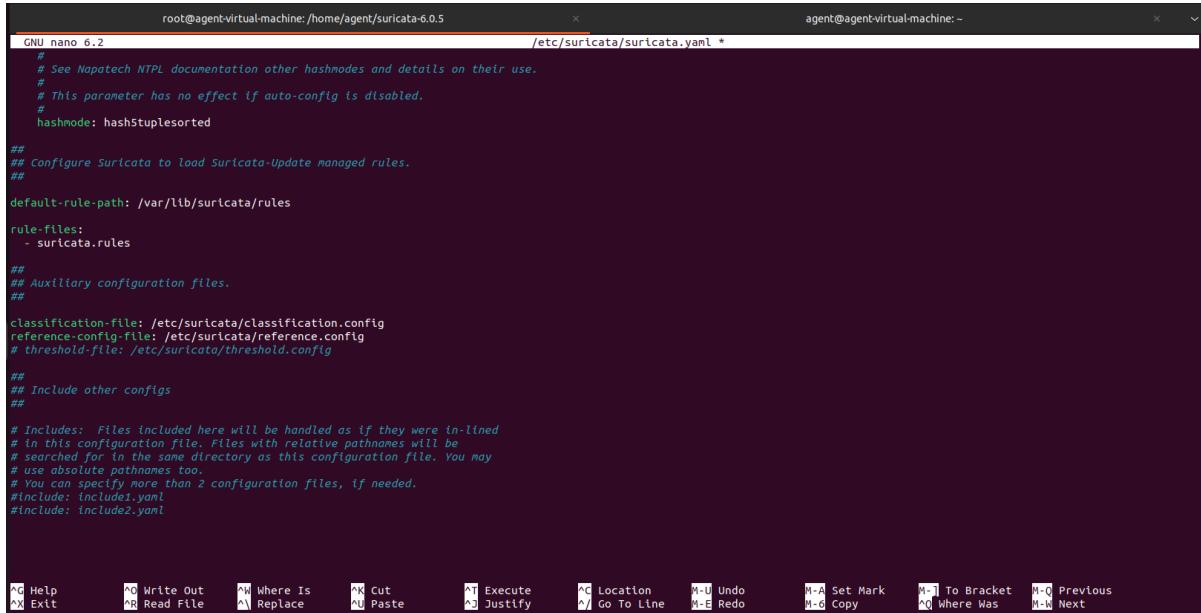
```
# nano /etc/suricata/suricata.yaml
```

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.



```
root@agent-virtual-machine:/home/agent/suricata-6.0.5          agent@agent-virtual-machine: ~
GNU nano 6.2                                     /etc/suricata/suricata.yaml *
#
# See Napatech NTPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hashStuplesorted

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

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

rule-files:
- suricata.rules

##
## Auxiliary configuration files.
##

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

##
## Include other configs
##

# Includes: Files included here will be handled as if they were in-lined
# in this configuration file. Files with relative pathnames will be
# searched for in the same directory as this configuration file. You may
# use absolute pathnames too.
# You can specify more than 2 configuration files, if needed.
#include: include1.yaml
#include: include2.yaml

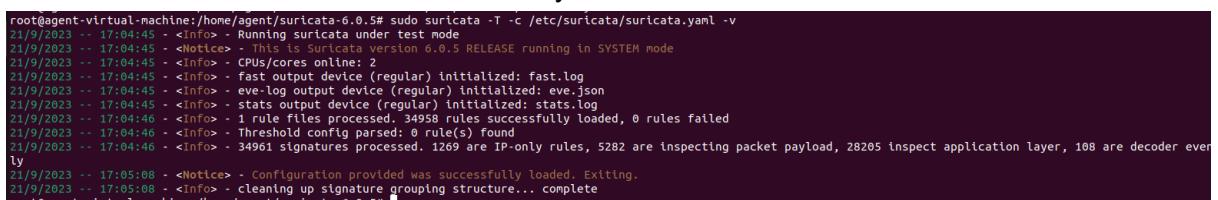
^C Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute     ^C Location    ^U Undo       ^A Set Mark   ^J To Bracket  ^O Previous
^X Exit      ^R Read File   ^H Replace    ^P Paste       ^G Justify     ^I Go To Line  ^E Redo       ^B Copy       ^L Where Was   ^K Next
```

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
```



```
root@agent-virtual-machine:/home/agent/suricata-6.0.5# sudo suricata -T -c /etc/suricata/suricata.yaml -v
21/9/2023 -- 17:04:45 - <Info> - Running suricata under test mode
21/9/2023 -- 17:04:45 - <Notice> - This is Suricata version 6.0.5 RELEASE running in SYSTEM mode
21/9/2023 -- 17:04:45 - <Info> - CPUs/cores online: 2
21/9/2023 -- 17:04:45 - <Info> - fast output device (regular) initialized: fast.log
21/9/2023 -- 17:04:45 - <Info> - eve-log output device (regular) initialized: eve.json
21/9/2023 -- 17:04:45 - <Info> - stats output device (regular) initialized: stats.log
21/9/2023 -- 17:04:46 - <Info> - 1 rule file processed, 34958 rules successfully loaded, 0 rules failed
21/9/2023 -- 17:04:46 - <Info> - Threshold config parsed: 0 rule(s) found
21/9/2023 -- 17:04:46 - <Info> - 34961 signatures processed. 1269 are IP-only rules, 5282 are inspecting packet payload, 28205 inspect application layer, 108 are decoder every
21/9/2023 -- 17:05:08 - <Notice> - Configuration provided was successfully loaded. Exiting.
21/9/2023 -- 17:05:08 - <Info> - cleaning up signature grouping structure... complete
21/9/2023 -- 17:05:08 - <Info> - exiting with code 0
```

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>
```

The screenshot shows a terminal window with the nano text editor open. The file being edited is /var/ossec/etc/ossec.conf. The configuration includes a new section for collecting Suricata logs:

```
GNU nano 6.2
[...]
<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 spot the events related to network traffic on the respective Suricata server agent events.

See sample screenshot below;

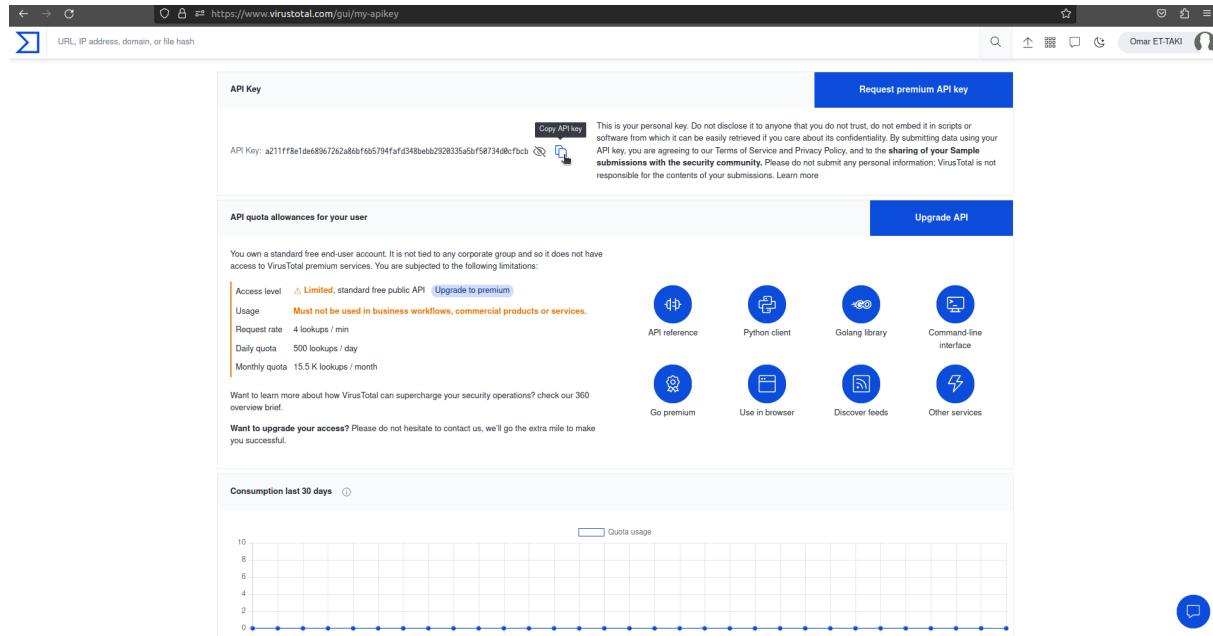
Time	Technique(s)	Tactic(s)	Description	Level	Rule ID
Sep 21, 2023 @ 17:37:39.703			Agent event queue is flooded. Check the agent configuration.	12	204
Sep 21, 2023 @ 17:37:25.016			Agent event queue is full. Events may be lost.	9	203
Sep 21, 2023 @ 17:37:23.289			Agent event queue is full. Events may be lost.	9	203
Sep 21, 2023 @ 17:37:23.259			Agent event queue is 90% full.	7	202
Sep 21, 2023 @ 17:36:07.925			PAM: Login session closed.	3	5502
Sep 21, 2023 @ 17:36:04.327			Host-based anomaly detection event (rootcheck).	7	510
Sep 21, 2023 @ 17:36:04.311			Host-based anomaly detection event (rootcheck).	7	510
Sep 21, 2023 @ 17:36:03.371			Wazuh agent started.	3	503
Sep 21, 2023 @ 12:30:17.960			SCA summary: CIS Ubuntu Linux 22.04 LTS Benchmark v1.0.0: Score less than 50% (39)	7	19004

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.

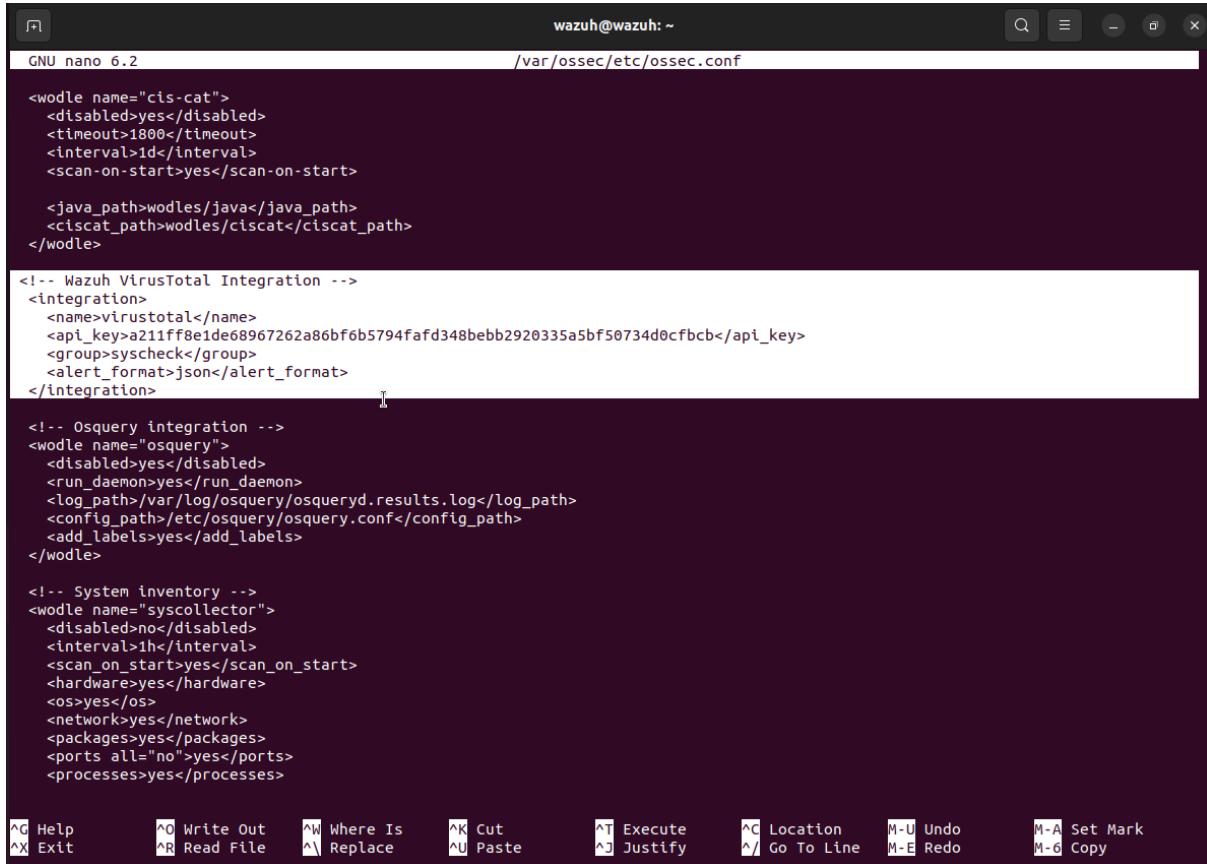
Once you have created an account and logged into VirusTotal, click on your username at the top right corner > API Key.



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.



```
wazuh@wazuh: ~
GNU nano 6.2          /var/ossec/etc/ossec.conf

<wodle name="cis-cat">
  <disabled>yes</disabled>
  <timeout>1800</timeout>
  <interval>1d</interval>
  <scan-on-start>yes</scan-on-start>

  <java_path>wodles/java</java_path>
  <ciscat_path>wodles/ciscat</ciscat_path>
</wodle>

<!-- Wazuh VirusTotal Integration -->
<integration>
  <name>virustotal</name>
  <api_key>a211ff8e1de68967262a86bf6b5794fafd348bebb2920335a5bf50734d0cfbcb</api_key>
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>

<!-- Osquery integration -->
<wodle name="osquery">
  <disabled>yes</disabled>
  <run_daemon>yes</run_daemon>
  <log_path>/var/log/osquery/osqueryd.results.log</log_path>
  <config_path>/etc/osquery/osquery.conf</config_path>
  <add_labels>yes</add_labels>
</wodle>

<!-- System inventory -->
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>1h</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>
```

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.

The screenshot shows the Wazuh dashboard with the 'Modules' tab selected. In the left sidebar, 'Settings' is highlighted. The main area displays various monitoring and auditing metrics. Under 'AUDITING AND POLICY MONITORING', there are three sections: 'Policy monitoring', 'System auditing', and 'Security configuration assessment'. At the bottom, there are links for 'Vulnerabilities', 'VirusTotal', 'PCI DSS', and 'NIST 800-53'.

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

The screenshot shows the 'Settings' page under the 'wazuh' tab. The 'Threat Detection and Response' section is expanded, showing several modules: 'Vulnerabilities' (disabled), 'MITRE ATT&CK' (disabled), 'VirusTotal' (enabled with a blue switch icon), 'Osquery' (disabled), and 'Docker listener' (disabled). Below this is the 'Regulatory Compliance' section, which includes 'PCI DSS' (disabled) and 'NIST 800-53' (disabled).

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.

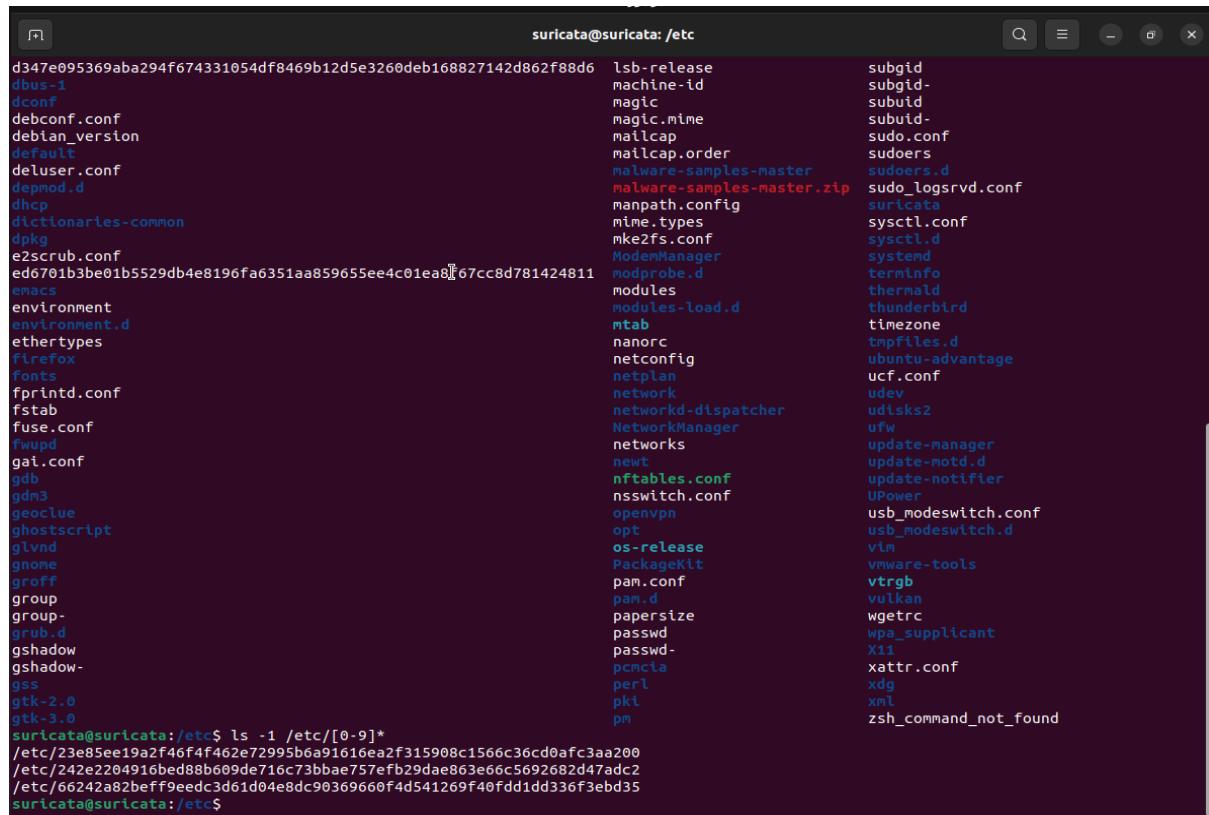
Demonstrating how Wazuh monitors the directories for Malicious Files

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.

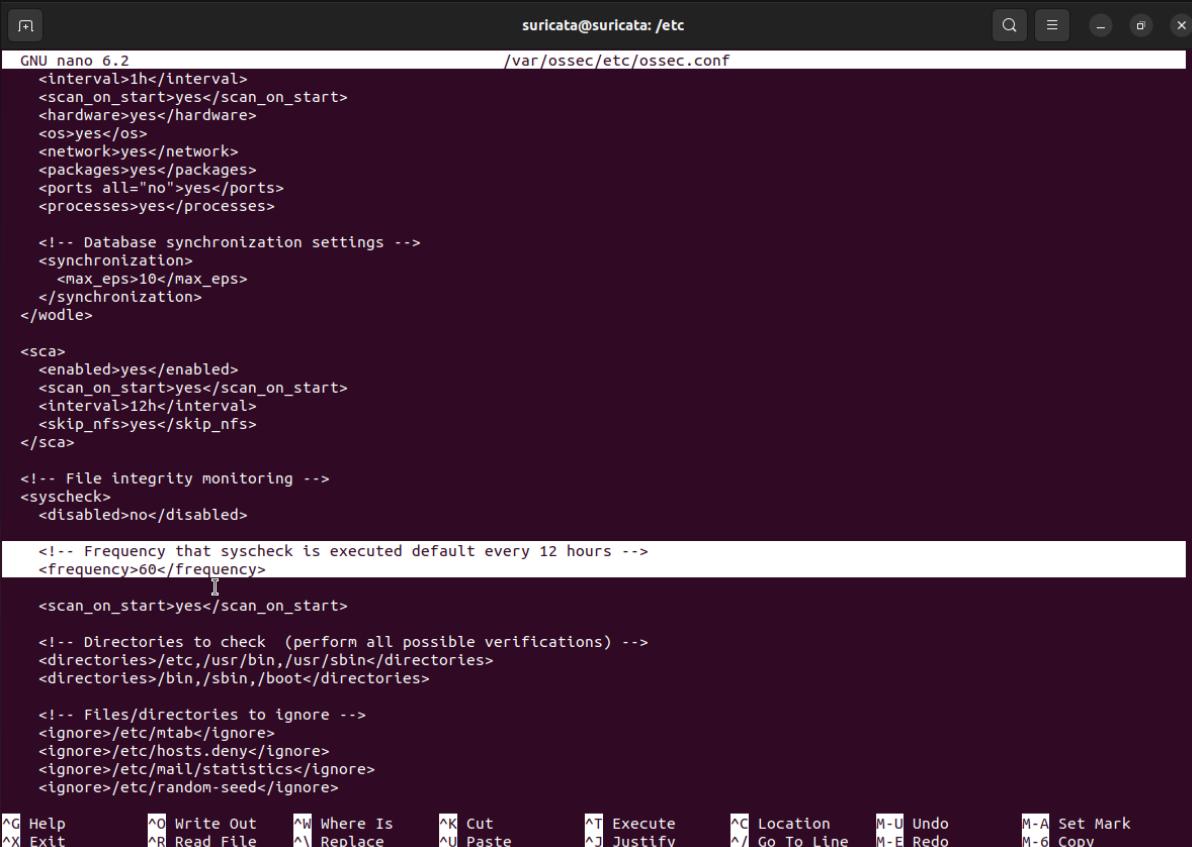
NOTE: This is done on an isolated temporary test system. Do not place malicious files on your systems!!!! We cannot be held responsible on what malicious files may do to your system.

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;



```
suricata@suricata: /etc
d347e095369aba294f674331054df8469b12d5e3260deb168827142d862f88d6  lsb-release          subgid
dbus-1                machine-id        subgid-
dconf                magic              subuid
debconf.conf          magic.mime        subuid-
debian_version        mailcap            sudo.conf
default               mailcap.order      sudoers
deluser.conf          malware-samples-master sudoers.d
depmod.d              malware-samples-master.zip sudo_logsrvd.conf
dhcp                 manpath.config    suricata
dictionaries-common   mime.types        sysctl.conf
dpkg                 mke2fs.conf       sysctl.d
e2scrub.conf          ModemManager     systemd
ed6701b3be01b5529db4e8196fa6351aa859655ee4c01ea8 67cc8d781424811  terminfo
enacs                modules           thermald
environment          modules-load.d   thunderbird
environment.d         mtab              timezone
ethertypes           nanorc            tmpfiles.d
firefox              netconfig          ubuntu Advantage
fonts                netplan            ucf.conf
fprintd.conf          network            udev
fstab                networkd-dispatcher NetworkManager
fuse.conf            NetworkManager   udisks2
fwupd                networks           ufw
gai.conf              neutw              update-manager
gdb                  nftables.conf     update-motd.d
gdm3                nsswitch.conf    update-notifier
geoclue              openvpn            UPower
ghostscript          opt                usb_modeswitch.conf
glvnd                os-release         usb_modeswitch.d
gnome               PackageKit        vim
grff                pan.conf          VMware-tools
group               pan.d              vtrgb
group-               papersize         vulkan
grub.d              passwd            wgetrc
gshadow              passwd-           wpa_supplicant
gshadow-             pcmcia            X11
gss                 perl              xattr.conf
gtk-2.0              pki               xdg
gtk-3.0              pm                xml
suricata@suricata:/etc$ ls -1 /etc/[0-9]*
/etc/23e85ee19a2f46f4f462e72995b6a91616ea2f315908c1566c36cd0afc3aa200
/etc/242e2204916bed88b609de716c73bbae757efb29dae863e66c5692682d47adc2
/etc/66242a82beff9eedc3d61d04e8dc90369660f4d541269f40fdd1dd336f3ebd35
suricata@suricata:/etc$
```

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



```
suricata@suricata: /etc
GNU nano 6.2                               /var/ossec/etc/ossec.conf

<interval>1h</interval>
<scan_on_start>yes</scan_on_start>
<hardware>yes</hardware>
<os>yes</os>
<network>yes</network>
<packages>yes</packages>
<ports all="no">yes</ports>
<processes>yes</processes>

<!-- Database synchronization settings -->
<synchronization>
  <max_eps>10</max_eps>
</synchronization>
</wodle>

<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>

<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>

<!-- Frequency that syscheck is executed default every 12 hours -->
<frequency>60</frequency>
  |
<scan_on_start>yes</scan_on_start>

<!-- Directories to check (perform all possible verifications) -->
<directories>/etc,/usr/bin,/usr/sbin</directories>
<directories>/bin,/sbin,/boot</directories>

<!-- Files/directories to ignore -->
<ignore>/etc/mtab</ignore>
<ignore>/etc/hosts.deny</ignore>
<ignore>/etc/mail/statistics</ignore>
<ignore>/etc/random-seed</ignore>
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^L Go To Line M-E Redo M-C Copy

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.

The screenshot shows the Wazuh Settings interface. On the left, there's a sidebar with categories like Management, Agents, Tools, Security, and Settings. Under Settings, there are sections for Modules, Security information management, Threat detection and response, and Regulatory Compliance. In the Threat detection and response section, 'VirusTotal' is highlighted with a black box. Below this, there are three toggle switches for GDPR, HIPAA, and TSC, all of which are checked.

You can click Events to see the related events.

The screenshot shows the Wazuh Overview interface. At the top, it says 'Dashboard Events'. Below that is a search bar with filters: 'manager.name: wazuh' and 'rule.groups: virustotal'. There are also buttons for 'KQL', 'Last 24 hours', 'Show dates', and 'Refresh'. The main area has three donut charts: 'Total malicious' (4), 'Total positives' (8), and 'Total' (8). Below these are two pie charts: 'Unique malicious files per agent' (suricata) and 'Last scanned files' (with four segments: green, purple, red, blue). A table on the right lists 'Last scanned files' with their file paths and sizes.

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.

The screenshot shows a browser window titled "Wazuh - Elastic" with the URL [https://192.168.59.138/app/wazuh#/overview/?tab=virustotal&_g=\(filters:!\(\),refreshInterval:\(pause](https://192.168.59.138/app/wazuh#/overview/?tab=virustotal&_g=(filters:!(),refreshInterval:(pause). The page displays a list of security events under the "VirusTotal" tab. On the left, there is a sidebar with a tree view of event types, including "agent.ip", "data.integration", "data.virustotal.description", "data.virustotal.error", "data.virustotal.found", "data.virustotal.scan_date", "data.virustotal.sha1", "data.virustotal.source.alert_id", "data.virustotal.source.md5", "data.virustotal.source.sha1", "decoder.name", "id", "input.type", "location", "manager.name", "rule.description", "rule.firetimes", "rule.gdpr", and "rule.groups". The main content area shows four events:

Date	Source	File Path	URL	Count
Jul 29, 2023 @ 15:32:32.343	wazuh	-	-	-
Jul 29, 2023 @ 15:32:31.238	suricata	/etc/ed6701b3be01b5529db4e 8196fa6351aa859655ee4c01ea 8f67cc8d781424811	https://www.virustotal.com/gui/file/ed6701b3be01b5529db4e8196fa6351aa859655ee4c01ea8f67cc8d781424811/detection/f-ed6701b3be01b5529db4e8196fa6351aa859655ee4c01ea8f67cc8d781424811-1660	1
Jul 29, 2023 @ 15:32:29.904	suricata	/etc/242e2204916bed88b609de716c73bbae757efb29daea863e 66c569262d47adc2	https://www.virustotal.com/gui/file/242e2204916bed88b609de716c73bbae757efb29daea863e66c569262d47adc2/detection/f-242e2204916bed88b609de716c73bbae757efb29daea863e66c569260d7adec2-1660	1
Jul 29, 2023 @ 15:32:28.366	suricata	/etc/66242a82beff9eedc d04e8dc90369660f4d541269f4 0fd1dd336f3ebd35	https://www.virustotal.com/gui/file/66242a82beff9eedc3d61d04e8dc90369660f4d541269f40fd1dd336f3ebd35/detection/f-66242a82beff9eedc3d61d04e8dc90369660f4d541269f40fd1dd336f3ebd35-1660	1
Jul 29, 2023 @ 15:32:27.014	suricata	/etc/23e85ee19a2f46f4f462e 72995b6a91616ea2f315908c15 66c36cd0afc3aa200	https://www.virustotal.com/gui/file/23e85ee19a2f46f4f462e72995b6a91616ea2f315908c1566c36cd0afc3aa200/detection/f-23e85ee19a2f46f4f462e72995b6a91616ea2f315908c1566c36cd0afc3aa200-1688	1

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
```

```
root@agent-virtual-machine:/home/agent# sudo apt-get update && sudo apt-get install python3
Hit:1 https://packages.wazuh.com/4.x/apt stable InRelease
Hit:2 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.0 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [785 kB]
Get:7 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [559 kB]
Get:9 http://ma.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [493 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [143 kB]
Get:11 http://ma.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,012 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [40.1 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.7 kB]
Get:14 http://ma.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [227 kB]
Get:15 http://ma.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [101 kB]
Get:16 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [655 kB]
Get:17 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [984 kB]
Get:18 http://ma.archive.ubuntu.com/ubuntu jammy-updates/universe/amd64 DEP-11 Metadata [289 kB]
Get:19 http://ma.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:20 http://ma.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [64.3 kB]
Get:21 http://ma.archive.ubuntu.com/ubuntu jammy-backports/main i386 Packages [56.5 kB]
Get:22 http://ma.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [4,916 B]
Get:23 http://ma.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://ma.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [16.9 kB]
Get:25 http://ma.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.8 kB]
Get:26 http://ma.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.8 kB]
Get:27 http://ma.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [640 B]
Fetched 5,875 kB in 6s (1,034 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@agent-virtual-machine:/home/agent# sudo apt-get install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

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-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/libblas/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)
    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=5c6f4944684f68e42cfabf0553f911280f265e0394c427ee1a692584e152a88
    Stored in directory: /root/.cache/pip/wheels/da/bd/0342ac886d4deb166a3191eb2566f738c5b1574cb0a8cd62
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 its 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
        json_output['nmap_port_name'] = nm[host][proto][port]
        json_output['nmap_port_state'] = nm[host][proto][port].state()
        json_output['nmap_port_service'] = nm[host][proto][port].service()

    results.append(json_output)

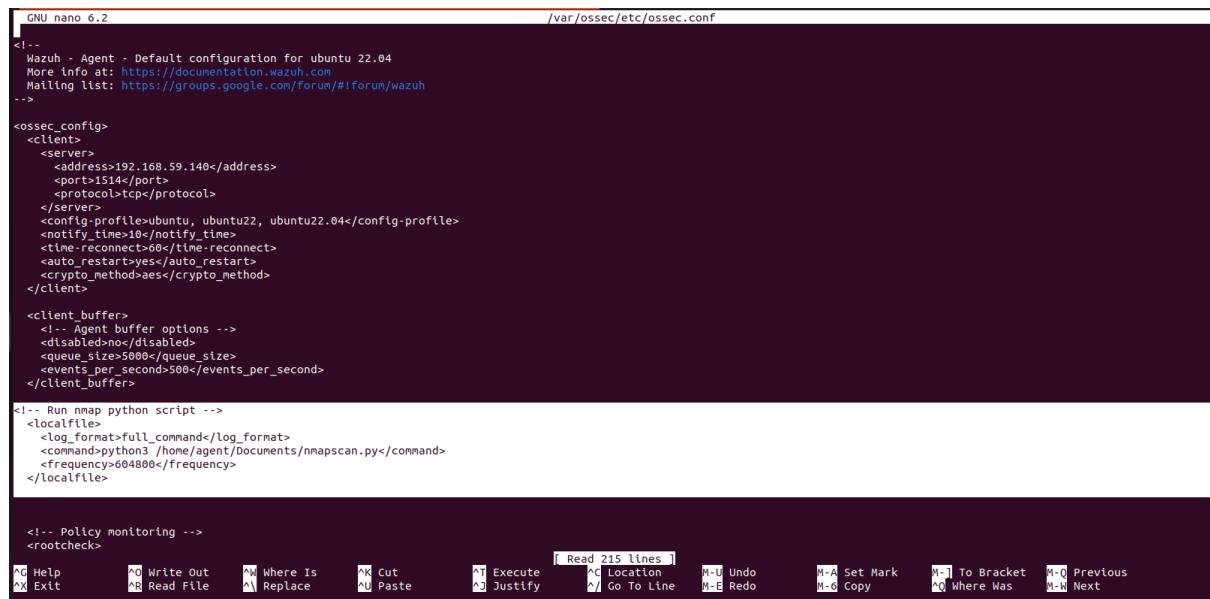
    return results
```

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.



```
GNU nano 6.2                               /var/ossec/etc/ossec.conf
[...]
<!-- Run nmap python script -->
<localfile>
<log_format>full_command</log_format>
<command>python3 /home/<USERNAME>/Documents/nmapscan.py</command>
<frequency>604800</frequency>
</localfile>
[...]
```

3. Restart the Wazuh agent to apply this change:

```
# sudo systemctl restart wazuh-agent
```

Wazuh server :

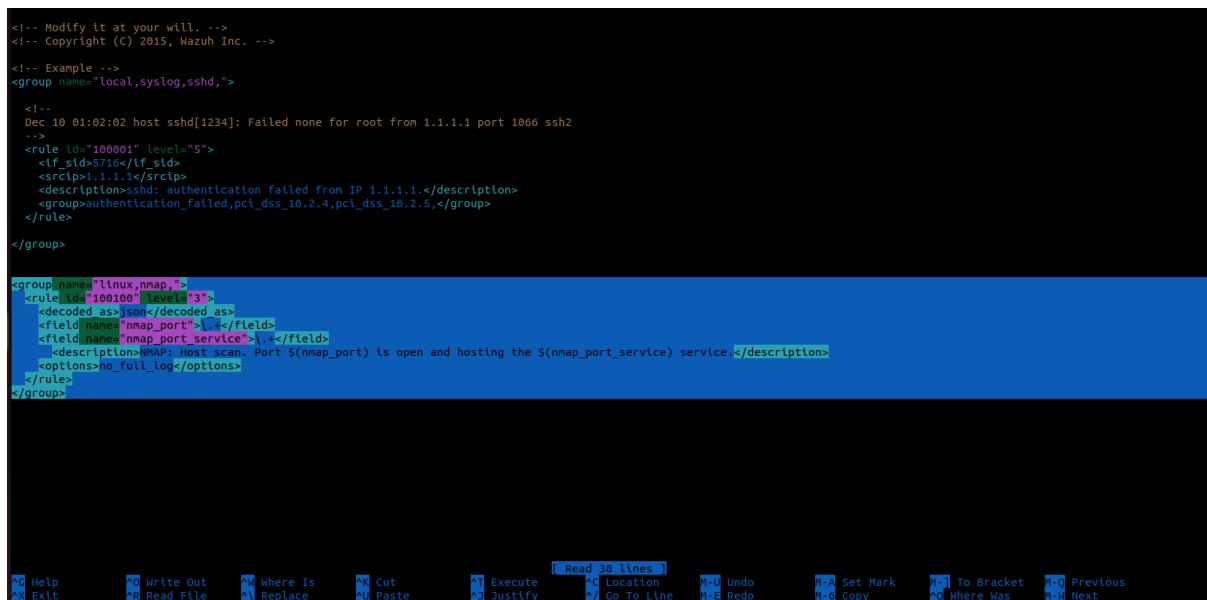
In this section, we create a rule to capture the result of the Nmap scan on the monitored endpoint.

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.



```
<!-- Modify it at your will. -->
<!-- Copyright (C) 2015, Wazuh Inc. -->
<!-- Example -->
<group name="local,syslog,sshd,">
  <!--
  Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
  -->
  <rule id="100001" level="5">
    <if_sid>5710</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>ssh: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>
</group>

<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>
```

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.

