

INTEGRATION OF WAZUH WITH MODSECURITY AND FAIL2BAN

Introduction

This document provides a detailed, step-by-step guide to installing and integrating the Wazuh Security Information and Event Management (SIEM) system with OpenSearch and Filebeat on an Ubuntu server. The aim is to build a centralized security monitoring platform that collects, analyzes, and visualizes security events from multiple sources in real time.

The main components involved in this setup are:

Wazuh Manager: This is the core of the Wazuh system. It receives security data from multiple agents, analyzes it using predefined rules, detects threats, and generates alerts.

Wazuh Agent: Installed on monitored systems (in this case, a Kali Linux machine), the agent collects logs and system information, including data from security tools like ModSecurity and Fail2Ban. It then forwards this data securely to the Wazuh Manager.

OpenSearch: This is a search and analytics engine that stores and indexes the collected log data, allowing for fast querying and data retrieval.

Wazuh Dashboard: A user-friendly web interface that allows administrators and security analysts to visualize alerts, monitor the security status of systems, and manage Wazuh configurations.

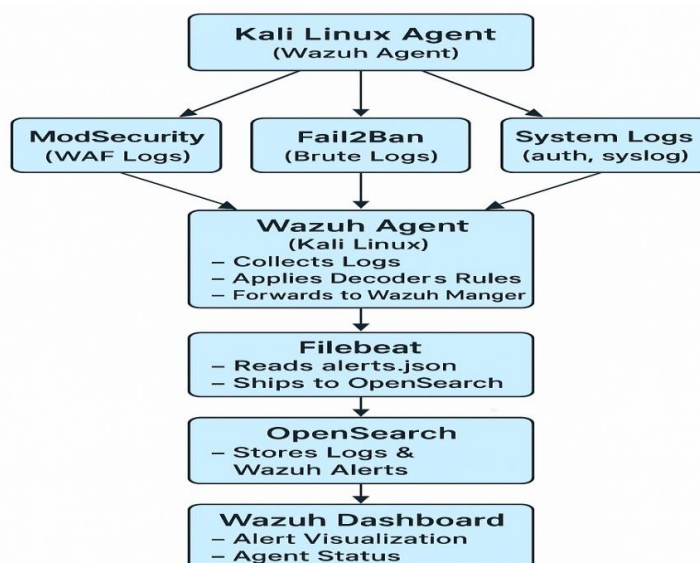
Filebeat: A lightweight log shipper installed on the manager server, which forwards Wazuh logs and alerts to OpenSearch for indexing and storage.

In this project, the Kali Linux machine runs an Nginx server protected by ModSecurity (a web application firewall) and Fail2Ban (an intrusion prevention system). The logs from these security tools are collected by the Wazuh agent and sent to the Wazuh Manager, enabling comprehensive security monitoring and real-time threat detection.

This report will guide you through the entire installation and integration process, providing clear explanations, configurations, and screenshots, helping both system administrators and cybersecurity professionals deploy a fully functional and effective SIEM solution.

Overview of Wazuh Manager, Agents, and Supporting Components

This setup includes several components working together to provide effective security monitoring. In this architecture review, Wazuh agents installed on endpoints collect security data and send it to the Wazuh manager, which analyzes the information. Filebeat ships log data to OpenSearch, where it is stored and indexed, while the Wazuh Dashboard offers an intuitive interface for visualization and alert management.



Part1: Installing Wazuh Manager, wazuh indexer, wazuh dashboard, Filebeat on Ubuntu

installing and Configuring Wazuh Indexer Step by Step

Prerequisites

Root or sudo privileges on all nodes
Network connectivity between Wazuh components
Basic knowledge of Linux command line

Stage 1: Certificates Creation

1. Purpose

Create SSL certificates to encrypt communications between Wazuh components (indexer, manager, dashboard).

`curl -sO https://packages.wazuh.com/4.12/wazuh-certs-tool.sh`

`curl -sO https://packages.wazuh.com/4.12/config.yml`

2. Edit config.yml to define your cluster nodes with their names and IP addresses. Example snippet:

nodes:

indexer:

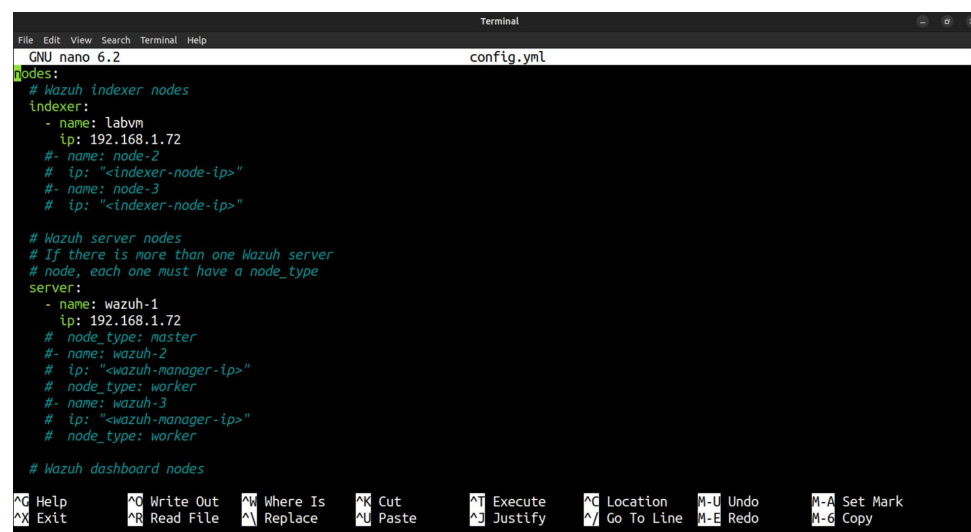
- name: node-1 #change the node-1 to wazuh indexer IP address
- ip: "<indexer-node-ip>"

server:

- name: wazuh-1
- ip: "<wazuh-manager-ip>"

dashboard:

- name: dashboard
- ip: "<dashboard-node-ip>"



```
GNU nano 6.2 config.yml
nodes:
# Wazuh indexer nodes
indexer:
- name: labvm
  ip: 192.168.1.72
#- name: node-2
# ip: "<indexer-node-ip>"
#- name: node-3
# ip: "<indexer-node-ip>"

# Wazuh server nodes
# If there is more than one Wazuh server
# node, each one must have a node_type
server:
- name: wazuh-1
  ip: 192.168.1.72
# node_type: master
#- name: wazuh-2
# ip: "<wazuh-manager-ip>"
# node_type: worker
#- name: wazuh-3
# ip: "<wazuh-manager-ip>"
# node_type: worker

# Wazuh dashboard nodes
```

3. Run the script to generate certificates:

`bash ./wazuh-certs-tool.sh -A`

4. Compress certificates for distribution:

`tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates/ .`
`rm -rf ./wazuh-certificates`

5. Copy wazuh-certificates.tar to all nodes (indexer, manager, dashboard) using scp or similar.

Stage 2: Nodes Installation

1. Install Dependencies

On each node, install required packages depending on your Linux distribution:

```
apt-get install debconf adduser procps gnupg apt-transport-https
```

2. Add Wazuh Repository

```
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
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
apt-get update
```

3. Install Wazuh Indexer Package

```
apt-get -y install wazuh-indexer
```

4. Configure Wazuh Indexer

Edit `/etc/wazuh-indexer/opensearch.yml` to reflect your cluster nodes:

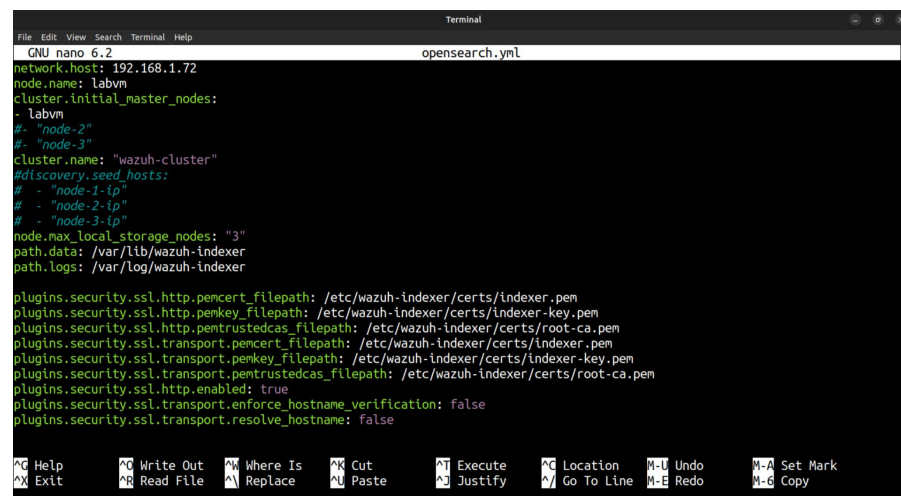
Set `network.host` to the node IP

Set `node.name` as per `config.yml`

Configure `cluster.initial_master_nodes` with master node names

Configure `discovery.seed_hosts` with master node IPs (for multi-node)

Set `plugins.security.nodes_dn` with certificate distinguished names



```
GNU nano 6.2 opensearch.yml
network.host: 192.168.1.72
node.name: labvm
cluster.initial_master_nodes:
- labvm
#- "node-2"
#- "node-3"
cluster.name: "wazuh-cluster"
#discovery.seed_hosts:
# - "node-1-ip"
# - "node-2-ip"
# - "node-3-ip"
node_max_local_storage_nodes: "3"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer

plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.transport.resolve_hostname: false
```

5. Deploy Certificates on Each Node On each indexer node:

```
NODE_NAME=<wazuh-indexer IP> #set wazuh-indexer to 192.168.1.72
```

```
mkdir /etc/wazuh-indexer/certs
```

```
tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/ ./192.168.1.72.pem ./192.168.1.72-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem
```

```
mv -n /etc/wazuh-indexer/certs/192.168.1.72.pem /etc/wazuh-indexer/certs/indexer.pem
```

```
mv -n /etc/wazuh-indexer/certs/192.168.1.72-key.pem /etc/wazuh-indexer/certs/indexer-key.pem
```

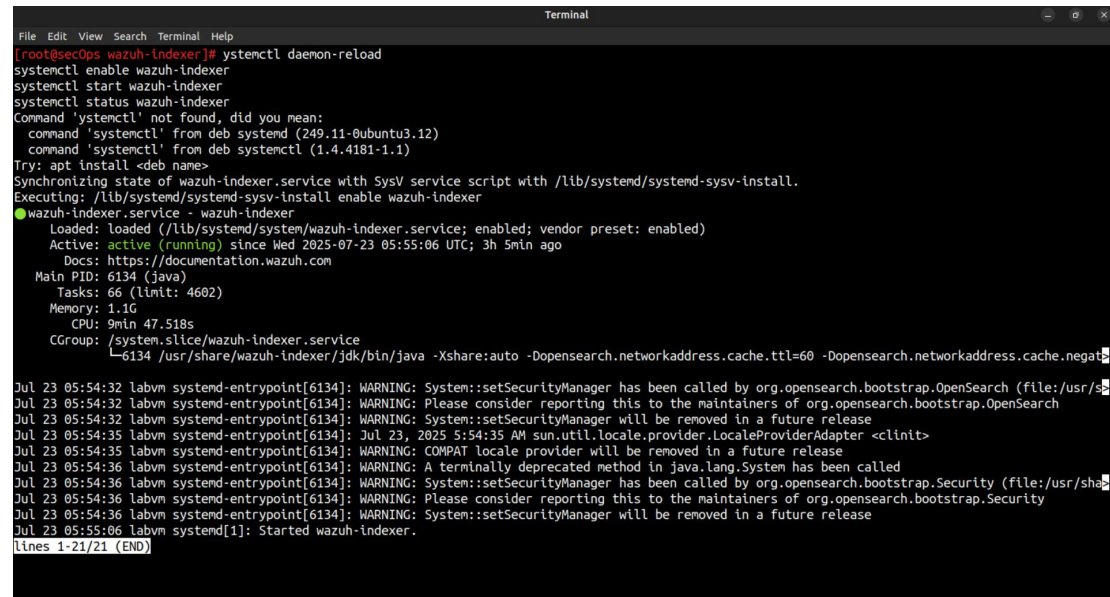
```
chmod 500 /etc/wazuh-indexer/certs
```

```
chmod 400 /etc/wazuh-indexer/certs/*
```

```
chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```

6. Start and Enable Wazuh Indexer Service

`systemctl daemon-reload`
`systemctl enable wazuh-indexer`
`systemctl start wazuh-indexer`
`systemctl status wazuh-indexer`



```
File Edit View Search Terminal Help
[root@secOps wazuh-indexer]# systemctl daemon-reload
systemctl enable wazuh-indexer
systemctl start wazuh-indexer
systemctl status wazuh-indexer
Command 'systemctl' not found, did you mean:
  command 'systemctl' from deb systemd (249.11-0ubuntu3.12)
  command 'systemctl' from deb systemctl (1.4.4181-1.1)
Try: apt install <deb name>
Synchronizing state of wazuh-indexer.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-indexer
● wazuh-indexer.service - wazuh-indexer
   Loaded: loaded (/lib/systemd/system/wazuh-indexer.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-07-23 05:55:06 UTC; 3h 5min ago
     Docs: https://documentation.wazuh.com
   Main PID: 6134 (java)
    Tasks: 66 (limit: 4602)
   Memory: 1.1G
      CPU: 9min 47.518s
   CGroup: /system.slice/wazuh-indexer.service
           └─6134 /usr/share/wazuh-indexer/jdk/bin/java -Xshare:auto -Dopensearch.networkaddress.cache.ttl=60 -Dopensearch.networkaddress.cache.negative.ttl=10

Jul 23 05:54:32 labvm systemd-entrypoint[6134]: WARNING: System::setSecurityManager has been called by org.opensearch.bootstrap.OpenSearch (file:/usr/share/
Jul 23 05:54:32 labvm systemd-entrypoint[6134]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.OpenSearch
Jul 23 05:54:32 labvm systemd-entrypoint[6134]: WARNING: System::setSecurityManager will be removed in a future release
Jul 23 05:54:35 labvm systemd-entrypoint[6134]: Jul 23, 2025 5:54:35 AM sun.util.locale.provider.LocaleProviderAdapter <clinit>
Jul 23 05:54:35 labvm systemd-entrypoint[6134]: WARNING: COMPAT locale provider will be removed in a future release
Jul 23 05:54:36 labvm systemd-entrypoint[6134]: WARNING: A terminally deprecated method in java.lang.System has been called
Jul 23 05:54:36 labvm systemd-entrypoint[6134]: WARNING: System::setSecurityManager has been called by org.opensearch.bootstrap.Security (file:/usr/share/
Jul 23 05:54:36 labvm systemd-entrypoint[6134]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.Security
Jul 23 05:54:36 labvm systemd-entrypoint[6134]: WARNING: System::setSecurityManager will be removed in a future release
Jul 23 05:55:06 labvm systemd[1]: Started wazuh-indexer.
lines 1-21/21 (END)
```

7. Repeat these steps on each indexer node. Cluster Initialization On any indexer node, run:

`/usr/share/wazuh-indexer/bin/indexer-security-init.sh`

This initializes your single-node or multi-node cluster.

8. Testing the Installation

Test connection to the Wazuh Indexer API:

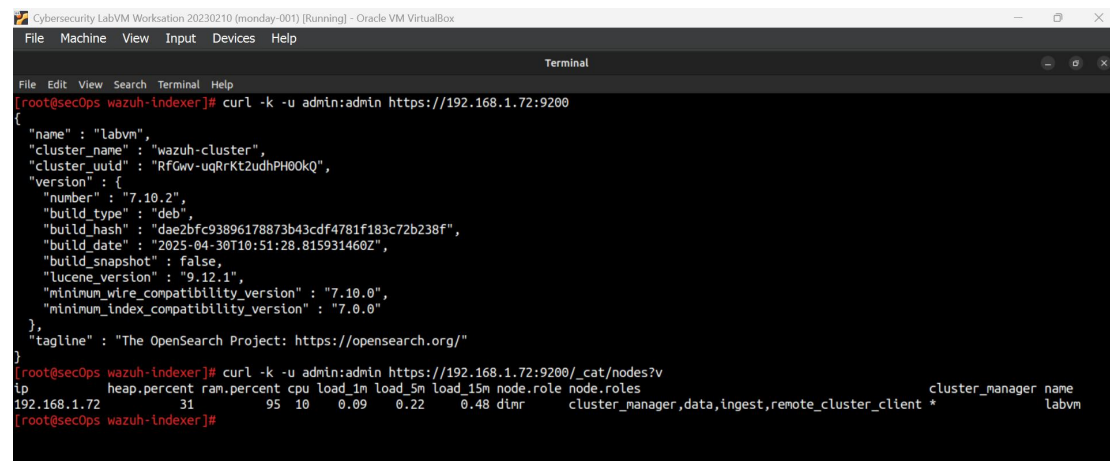
`curl -k -u admin:admin https://<WAZUH_INDEXER_IP_ADDRESS>:9200`

You should receive JSON output with cluster info.

Check cluster nodes:

`curl -k -u admin:admin https://<WAZUH_INDEXER_IP_ADDRESS>:9200/_cat/nodes?v`

I checked the manager's status to confirm it was active and running without errors.



```
Cybersecurity Lab/VM Workstation 20230210 (monday-001) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Edit View Search Terminal Help
[root@secOps wazuh-indexer]# curl -k -u admin:admin https://192.168.1.72:9200
{"name": "labvm",
 "cluster_name": "wazuh-cluster",
 "cluster_uuid": "RfGwv-uqRrKt2udhPH00kQ",
 "version": {
   "number": "7.10.2",
   "build_type": "deb",
   "build_hash": "dae2bfc93896178873b43cdf4781f183c72b238f",
   "build_date": "2025-04-30T10:51:28.815931460Z",
   "build_snapshot": false,
   "lucene_version": "9.12.1",
   "minimum_wire_compatibility_version": "7.10.0",
   "minimum_index_compatibility_version": "7.0.0"
 },
 "tagline": "The OpenSearch Project: https://opensearch.org/"
}
[root@secOps wazuh-indexer]# curl -k -u admin:admin https://192.168.1.72:9200/_cat/nodes?v
ip heap.percent ram.percent cpu load_1m load_5m load_15m node.role node.roles cluster_manager name
192.168.1.72 31 95 10 0.09 0.22 0.48 dlmr cluster_manager,data,ingest,remote_cluster_client * labvm
[root@secOps wazuh-indexer]#
```

Next, proceed with installing the Wazuh Server

step-by-step installation of a Wazuh server, including its core components:

Wazuh Manager: Collects and analyzes security data from agents.

Filebeat: Ships alerts and events securely to the Wazuh Indexer.

All steps require root privileges.

1. Update packages:

`apt-get update`

2. Install Core Components

Install Wazuh Manager:

`apt-get -y install wazuh-manager`

3. Update ossec.conf

Replace 0.0.0.0 with actual indexer IP:

```
<indexer>
<enabled>yes</enabled>
<hosts>
  <host>https://wazuh-ndexer-IP:9200</host>
</hosts>
<ssl>
  <certificate_authorities>
    <ca>/etc/filebeat/certs/root-ca.pem</ca>
  </certificate_authorities>
  <certificate>/etc/filebeat/certs/filebeat.pem</certificate>
  <key>/etc/filebeat/certs/filebeat-key.pem</key>
</ssl>
</indexer>
```

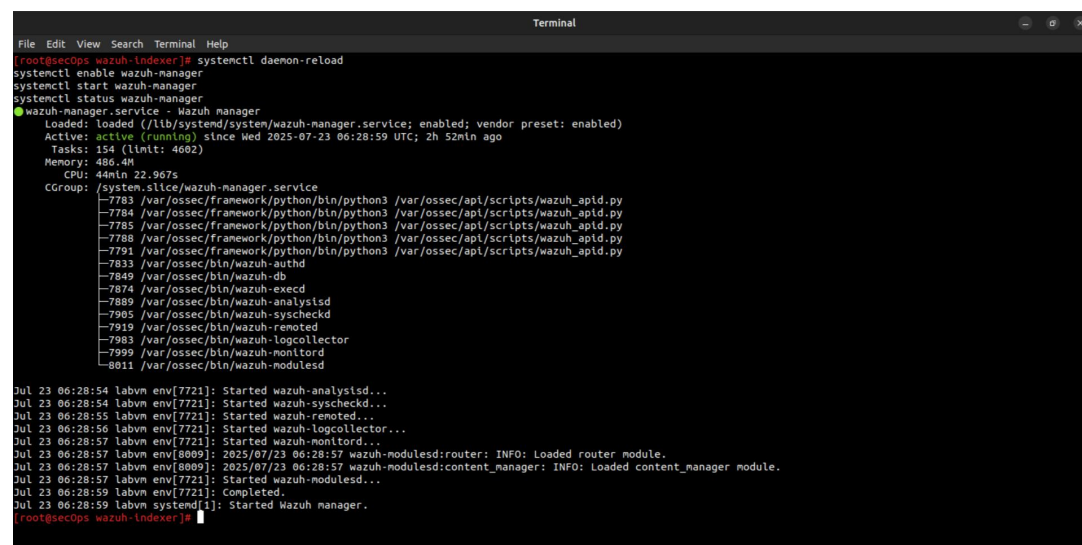
4. Start server

`systemctl daemon-reload`

`systemctl enable wazuh-manager`

`systemctl start wazuh-manager`

`systemctl status wazuh-manager`

A terminal window titled "Terminal" showing the installation and startup of Wazuh Manager. The user runs several systemctl commands: daemon-reload, enable, start, and status. The status command shows the service is loaded and active. Below this, the CGroup list is displayed, showing various Wazuh processes like wazuh-analysisd, wazuh-syscheckd, wazuh-remoted, wazuh-logcollector, wazuh-authd, wazuh-db, wazuh-execd, wazuh-modulesd, and wazuh-monitord. Finally, a series of log messages from the journal are shown, indicating the successful startup of these components and the Wazuh Manager itself.

```
File Edit View Search Terminal Help
[root@secOps wazuh-Indexer]# systemctl daemon-reload
systemctl enable wazuh-manager
systemctl start wazuh-manager
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 Wed 2025-07-23 06:28:59 UTC; 2h 52min ago
     Tasks: 154 (limit: 4602)
    Memory: 486.4M
       CPU: 44min 22.967s
   CGroup: /system.slice/wazuh-manager.service
           └─7783 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh_apid.py
             └─7784 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh_apid.py
               └─7785 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh_apid.py
                 └─7788 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh_apid.py
                   └─7791 /var/ossec/framework/python/bin/python3 /var/ossec/apl/scripts/wazuh_apid.py
                     └─7833 /var/ossec/bin/wazuh-authd
                       └─7849 /var/ossec/bin/wazuh-db
                         └─7874 /var/ossec/bin/wazuh-execd
                           └─7889 /var/ossec/bin/wazuh-analysisd
                             └─7905 /var/ossec/bin/wazuh-syscheckd
                               └─7919 /var/ossec/bin/wazuh-remoted
                                 └─7983 /var/ossec/bin/wazuh-logcollector
                                   └─7999 /var/ossec/bin/wazuh-monitord
                                     └─8011 /var/ossec/bin/wazuh-modulesd

Jul 23 06:28:54 labvm env[7721]: Started wazuh-analysisd...
Jul 23 06:28:54 labvm env[7721]: Started wazuh-syscheckd...
Jul 23 06:28:55 labvm env[7721]: Started wazuh-remoted...
Jul 23 06:28:56 labvm env[7721]: Started wazuh-logcollector...
Jul 23 06:28:57 labvm env[7721]: Started wazuh-monitord...
Jul 23 06:28:57 labvm env[8009]: 2025/07/23 06:28:57 wazuh-modulesd:router: INFO: Loaded router module.
Jul 23 06:28:57 labvm env[8009]: 2025/07/23 06:28:57 wazuh-modulesd:content_manager: INFO: Loaded content_manager module.
Jul 23 06:28:57 labvm env[7721]: Started wazuh-modulesd...
Jul 23 06:28:59 labvm env[7721]: Completed.
Jul 23 06:28:59 labvm systemd[1]: Started Wazuh manager.
[root@secOps wazuh-Indexer]#
```

Configuring Filebeat

1. Download Configuration

```
curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/4.12/tpl/wazuh/filebeat/filebeat.yml
```

2. Edit filebeat.yml → Set indexer nodes:

```
output.elasticsearch:
  hosts: ["192.168.1.72:9200"]
  protocol: https
  username: ${username}
  password: ${password}
```

3. Setup Keystore Credentials

```
filebeat keystore create
echo admin | filebeat keystore add username --stdin --force
echo admin | filebeat keystore add password --stdin --force #you can change password
```

4. Configure Alerts Template

```
curl -so /etc/filebeat/wazuh-template.json
https://raw.githubusercontent.com/wazuh/wazuh/v4.12.0/extensions/elasticsearch/7.x/wazuh-template.json
```

5. Install Wazuh Filebeat Module

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

6. deploy Certificates

Recall wazuh-certificates.tar exists:

```
NODE_NAME=<SERVER_NODE_NAME> #set node_name to wazuh-1
mkdir /etc/filebeat/certs
tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ ./wazuh-1.pem ./wazuh-1-key.pem ./root-ca.pem
```

```
mv -n /etc/filebeat/certs/wazuh-1.pem /etc/filebeat/certs/filebeat.pem
mv -n /etc/filebeat/certs/wazuh-key.pem /etc/filebeat/certs/filebeat-key.pem
```

```
chmod 500 /etc/filebeat/certs
chmod 400 /etc/filebeat/certs/*
chown -R root:root /etc/filebeat/certs
```

7. Configure Wazuh Indexer Connection

Step 4.1: Add credentials to Wazuh keystore

```
echo '<INDEXER_USERNAME>' | /var/ossec/bin/wazuh-keystore -f indexer -k username
echo '<INDEXER_PASSWORD>' | /var/ossec/bin/wazuh-keystore -f indexer -k password
```

8. Start filebeat

```
systemctl daemon-reload
systemctl enable filebeat
systemctl start filebeat
systemctl status filebeat
filebeat test output
```



```
Terminal
File Edit View Search Terminal Help
[root@secOps wazuh-indexer]# systemctl daemon-reload
systemctl enable filebeat
systemctl start filebeat
systemctl status filebeat
filebeat test output
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable 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 Wed 2025-07-23 05:55:14 UTC; 3h 49min ago
     Docs: https://www.elastic.co/products/beats/filebeat
    Main PID: 6938 (filebeat)
      Tasks: 8 (limit: 4602)
     Memory: 27.7M
        CPU: 15.481s
    CGroup: /system.slice/filebeat.service
            └─6938 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.hone /usr/share/filebeat --path.config /etc/filebeat --p

Jul 23 05:55:14 labvm systemd[1]: Started Filebeat sends log files to Logstash or directly to Elasticsearch..
[lines 1-12/12 (END)]
elasticsearch: https://192.168.1.72:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
  addresses: 192.168.1.72
  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: 7.10.2
[root@secOps wazuh-indexer]#
```

Next step is Wazuh Dashboard Installation

1. Prerequisites

Ensure you have root privileges.

Required tools: debhelper, tar, curl, libcap2-bin, gnupg, apt-transport-https.

`apt-get install debhelper tar curl libcap2-bin gnupg apt-transport-https`

2. Update repositories:

`apt-get update`

3. Install the Wazuh Dashboard

`apt-get -y install wazuh-dashboard`

4. Configure the Dashboard

Edit `/etc/wazuh-dashboard/opensearch_dashboards.yml` and update:

`server.host: 0.0.0.0`

`server.port: 443`

`opensearch.hosts: https://localhost:9200` #change localhost to wazuh indexer IP

`opensearch.ssl.verificationMode: certificate`

5. Deploy SSL Certificates

Replace `<DASHBOARD_NODE_NAME>` with your actual node name:

`NODE_NAME=dashboard`

`mkdir /etc/wazuh-dashboard/certs`

`tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ ./dashboard.pem ./dashboard-key.pem ./root-ca.pem`

`mv -n /etc/wazuh-dashboard/certsdashboard.pem /etc/wazuh-dashboard/certs/dashboard.pem`

`mv -n /etc/wazuh-dashboard/certs/dashboard-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem`

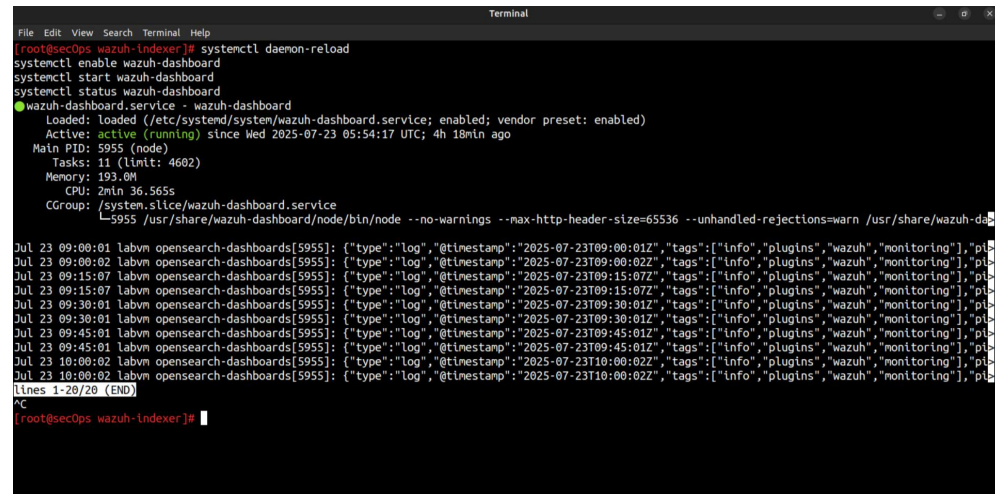
`chmod 500 /etc/wazuh-dashboard/certs`

`chmod 400 /etc/wazuh-dashboard/certs/*`

`chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs`

6. Start and Enable the Dashboard Service

```
systemctl daemon-reload
systemctl enable wazuh-dashboard
systemctl start wazuh-dashboard
systemctl status wazuh-dashboard
```



```
File Edit View Search Terminal Help
[root@secOps wazuh-indexer]# systemctl daemon-reload
systemctl enable wazuh-dashboard
systemctl start wazuh-dashboard
systemctl status wazuh-dashboard
● wazuh-dashboard.service - wazuh-dashboard
   Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-07-23 05:54:17 UTC; 4h 18min ago
     Main PID: 5955 (node)
       Tasks: 11 (limit: 4602)
      Memory: 193.0M
         CPU: 2min 36.565s
    CGroup: /system.slice/wazuh-dashboard.service
            └─5955 /usr/share/wazuh-dashboard/node/bin/node --no-warnings --max-http-header-size=65536 --unhandled-rejections=warn /usr/share/wazuh-da
Jul 23 09:00:01 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:00:01Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 09:00:02 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:00:02Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 09:15:07 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:15:07Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 09:15:07 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:15:07Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 09:30:01 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:30:01Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 09:30:01 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:30:01Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 09:45:01 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:45:01Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 09:45:01 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T09:45:01Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 10:00:02 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T10:00:02Z","tags":["info","plugins","wazuh","monitoring"],"pi
Jul 23 10:00:02 labvm opensearch-dashboards[5955]: {"type":"log","@timestamp":"2025-07-23T10:00:02Z","tags":["info","plugins","wazuh","monitoring"],"pi
lines 1-20/20 (END)
^C
[root@secOps wazuh-indexer]#
```

7. Connect Dashboard to Wazuh Server

Edit `/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml`:

hosts:

- default:

```
url: https://<WAZUH_SERVER_IP_ADDRESS>
port: 55000
username: wazuh-wui
password: wazuh-wui
run_as: false
```

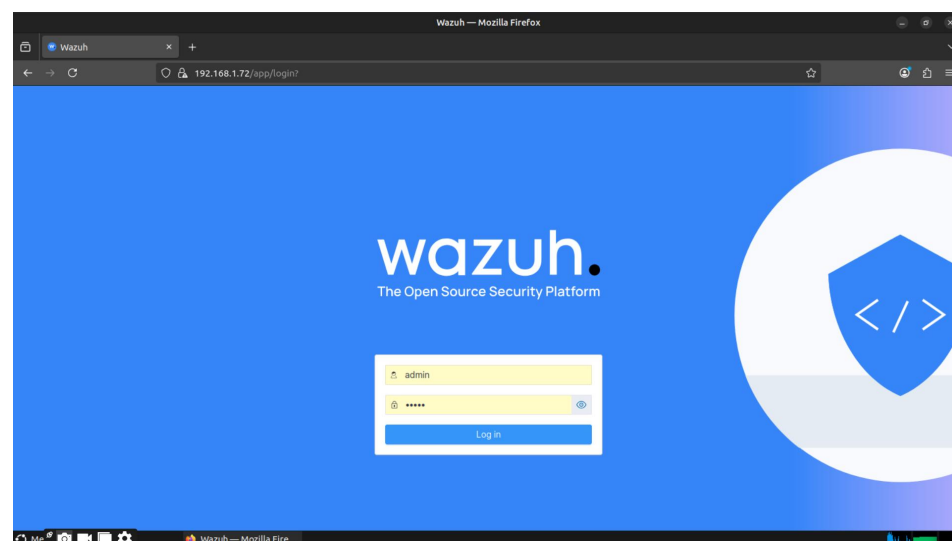
8. Access the Wazuh Dashboard

URL: `https://<WAZUH_DASHBOARD_IP_ADDRESS>`

Default credentials:

Username: admin

Password: admin



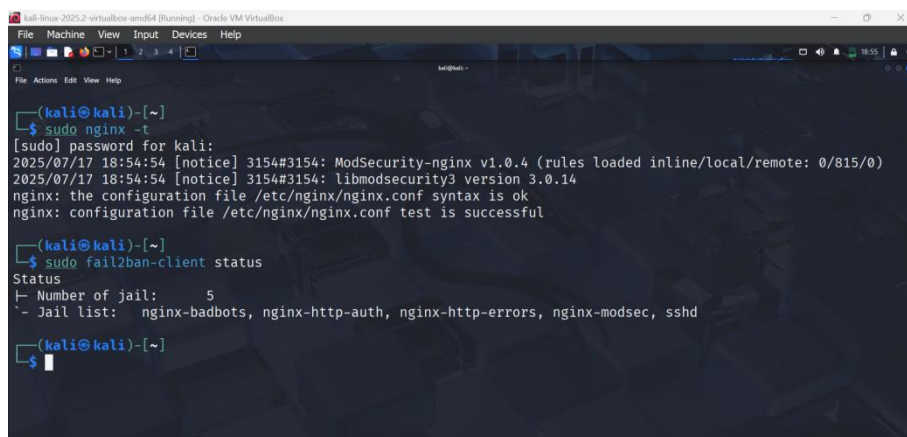
Part 2: Installing and Integrating Wazuh Agent on Kali Linux with ModSecurity and Fail2Ban

In this section, we extend the Wazuh setup by deploying the Wazuh Agent on Kali Linux, enabling it to monitor web application security events from ModSecurity and intrusion attempts blocked by Fail2Ban. This integration ensures that alerts generated by these tools are collected and forwarded to the Wazuh Manager for centralized analysis and visualization.

We break this part into three main steps:

1. Install and configure the Wazuh agent on Kali Linux to communicate with the Wazuh Manager.
2. Configure the agent to monitor log files generated by ModSecurity and Fail2Ban.
3. Ensure proper rule and decoder support on the Wazuh Manager to interpret and alert on relevant events.

This screenshots provide visual confirmation of the installation of ModSecurity and Fail2Ban, which are prerequisite tools installed before integrating Wazuh.



```
(kali@kali)-[~]
└─$ sudo nginx -t
[sudo] password for kali:
2025/07/17 18:54:54 [notice] 3154#3154: ModSecurity-nginx v1.0.4 (rules loaded inline/local/remote: 0/815/0)
2025/07/17 18:54:54 [notice] 3154#3154: libmodsecurity3 version 3.0.14
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

(kali@kali)-[~]
└─$ sudo fail2ban-client status
Status
- Number of jail:      5
- Jail list:  nginx-badbots, nginx-http-auth, nginx-http-errors, nginx-modsec, sshd

(kali@kali)-[~]
└─$
```

Step 1: Installing the Wazuh Agent on Kali Linux

To begin the integration process, we first installed the Wazuh agent on Kali Linux. This agent is responsible for collecting logs and security events from the local system, including those generated by ModSecurity and Fail2Ban.

1. Installation Steps:

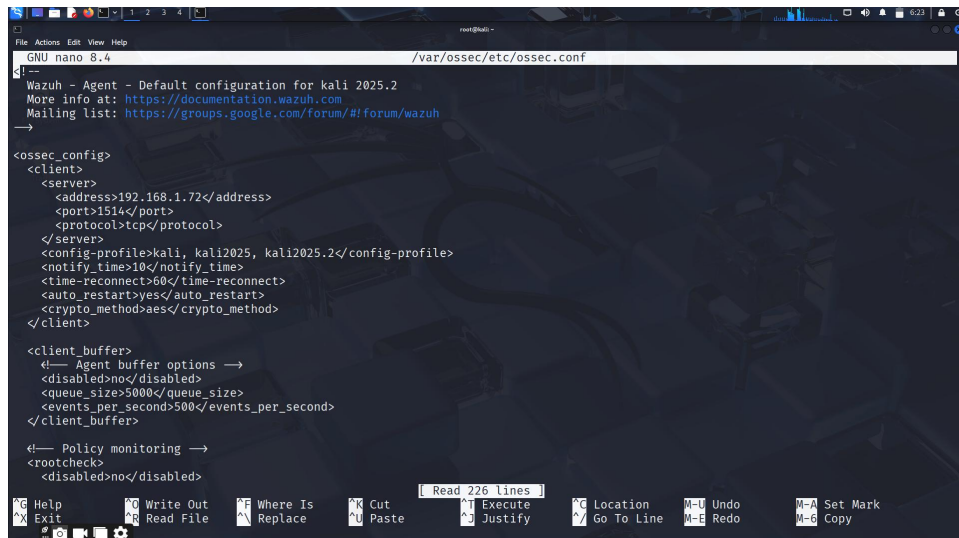
```
sudo apt update
sudo apt install wazuh-agent
```

Configure the Wazuh agent to communicate with the Wazuh manager:

#Edit the Wazuh agent configuration file:

```
sudo nano /var/ossec/etc/ossec.conf
#Replace or update the <server> section with your ubuntu IP address where wazuh manger is hosted.
For example my ubuntu server has IP: 192.168.43.177
```

```
<server>
  <address>your server IP</address>
  <port>1514</port>
  <protocol>udp</protocol>
</server>
```



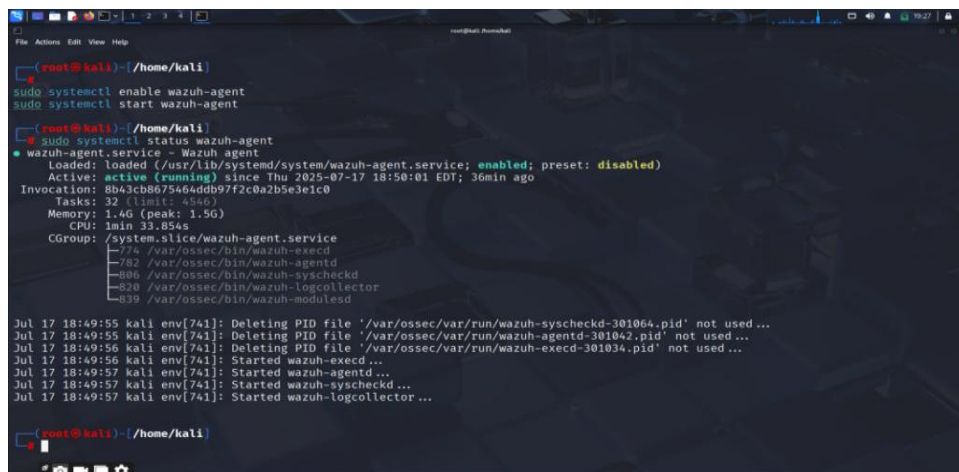
```
Wazuh - Agent - Default configuration for kali 2025.2
More info at: https://documentation.wazuh.com
Mailing list: https://groups.google.com/forum/#!forum/wazuh

<ossec_config>
  <client>
    <server>
      <address>192.168.1.72</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>kali, kali2025, kali2025.2</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
  </client>

  <client_buffer>
    <!-- Agent buffer options -->
    <disabled>no</disabled>
    <queue_size>5000</queue_size>
    <events_per_second>500</events_per_second>
  </client_buffer>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
  </rootcheck>
</ossec_config>
```

`sudo systemctl enable wazuh-agent`
`sudo systemctl start wazuh-agent`
`sudo systemctl status wazuh-agent`



```
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent

sudo systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; preset: disabled)
   Active: active (running) since Thu 2025-07-17 18:50:01 EDT; 36min ago
     Invocation: 8b43cb8675464ddb97f2c0a2b5e3e1c0
    Tasks: 32 (limit: 4546)
   Memory: 1.4G (peak: 1.5G)
      CPU: 1min 33.854s
   CGroup: /system.slice/wazuh-agent.service
           └─774 /var/ossec/bin/wazuh-execd
             └─782 /var/ossec/bin/wazuh-agentd
               └─806 /var/ossec/bin/wazuh-syscheckd
                 └─820 /var/ossec/bin/wazuh-logcollector
                   └─839 /var/ossec/bin/wazuh-modulesd

Jul 17 18:49:55 kali env[741]: Deleting PID file '/var/ossec/var/run/wazuh-syscheckd-301064.pid' not used ...
Jul 17 18:49:55 kali env[741]: Deleting PID file '/var/ossec/var/run/wazuh-agentd-301042.pid' not used ...
Jul 17 18:49:56 kali env[741]: Deleting PID file '/var/ossec/var/run/wazuh-execd-301034.pid' not used ...
Jul 17 18:49:56 kali env[741]: Started wazuh-execd ...
Jul 17 18:49:57 kali env[741]: Started wazuh-agentd ...
Jul 17 18:49:57 kali env[741]: Started wazuh-syscheckd ...
Jul 17 18:49:57 kali env[741]: Started wazuh-logcollector ...
```

Step 2: Configuring Wazuh to Monitor ModSecurity Logs and fail2ban

After installing the Wazuh agent, the next step is to configure it to monitor logs generated by ModSecurity and fail2ban. This enables Wazuh to detect and alert on suspicious web traffic patterns blocked or flagged by ModSecurity.

1. Ensure ModSecurity Audit Logging Is Enabled

Check the ModSecurity configuration file, usually located at `/etc/modsecurity/modsecurity.conf`, and verify the following directives:

`SecAuditEngine On`
`SecAuditLogRelevantStatus "^(?:5|4(?:?!04))"`
`SecAuditLog /var/log/modsec_audit.log`

These settings ensure that ModSecurity logs all relevant alerts into the specified log file.

2. Configure Wazuh to Read ModSecurity Logs

Edit the Wazuh agent configuration file:

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

#Add the following `<localfile>` section to monitor the ModSecurity audit log:

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/modsec_audit.log</location>
</localfile>
```

If your log format is syslog, use syslog or multi-line instead, depending on your setup.

4. Locate Fail2Ban Log File

By default, Fail2Ban writes logs to:

```
/var/log/fail2ban.log
```

You can confirm this in the Fail2Ban configuration:

```
sudo nano /etc/fail2ban/fail2ban.conf
```

Ensure the logtarget is set to the correct file:

```
logtarget = /var/log/fail2ban.log
```

5. Add Log Monitoring in Wazuh Agent Configuration

Open the Wazuh agent configuration file:

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

#Add the following block inside the <ossec_config> section:

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/fail2ban.log</location>
</localfile>
```

This tells Wazuh to monitor and parse entries in the Fail2Ban log.

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

Step3. Enable ModSecurity and fail2ban Rules in Wazuh make sure ModSecurity and fail2ban rules are enabled on your Wazuh manager. If you're using a local rules file:

```
sudo nano /var/ossec/etc/rules/local_rules.xml
```

#Add a basic rule (example):

```
<group name="modsecurity">
  <rule id="100001" level="5">
    <decoded_as>json</decoded_as>
    <field name="data.msg">ModSecurity</field>
    <description>ModSecurity alert detected</description>
  </rule>
</group>
```

2. Enable/Write Custom Rules for Fail2Ban (Optional)

If Fail2Ban is not already included in your Wazuh rules, you can create a basic rule:

```
/var/ossec/etc/decoders/local_decoder.xml:
<decoder name="fail2ban-decoder">
  <program_name>fail2ban</program_name>
</decoder>
```

```
sudo nano /var/ossec/etc/rules/local_rules.xml
```

```
<group name="fail2ban,">
  <rule id="100200" level="5">
    <decoded_as>eventlog</decoded_as>
```

```
<match>Ban</match>
<description>Fail2Ban banned an IP</description>
</rule>
</group>
```

1. ModSecurity Decoder and Rules

a) Create a Custom Decoder for ModSecurity

On the Wazuh Manager, create or edit the decoder file for ModSecurity:

```
sudo nano /var/ossec/etc/decoders/modsecurity_decoders.xml
```

#Add the following content to define a simple ModSecurity decoder:

```
<decoder name="modsecurity">
  <program_name>modsecurity</program_name>
  <type>audit</type>
</decoder>
```

#This decoder tells Wazuh to recognize and process logs coming from ModSecurity.

b) Create a Custom Rule for ModSecurity Alerts

Create or edit the rules file for ModSecurity:

```
sudo nano /var/ossec/etc/rules/modsecurity_rules.xml
```

#Add a basic rule to detect ModSecurity alerts:

```
<group name="modsecurity,">
  <rule id="100010" level="5">
    <decoded_as>modsecurity</decoded_as>
    <description>ModSecurity alert detected</description>
  </rule>
</group>
```

Note: This is a basic example. You can extend it later with more specific patterns to match different ModSecurity alerts.

2. Fail2Ban Decoder and Rules

a) Fail2Ban Basic Support

Wazuh includes basic Fail2Ban support by default, but you can enhance detection by adding or editing rules. Edit the local rules file:

```
sudo nano /var/ossec/etc/rules/local_rules.xml
```

Add a Fail2Ban detection rule like this:

```
<group name="fail2ban,">
  <rule id="100100" level="5">
    <match>fail2ban</match>
    <description>Fail2Ban alert detected</description>
  </rule>
</group>
```

3. Restart Wazuh Manager

After saving your decoder and rules files, restart the Wazuh manager service to apply changes:

```
sudo systemctl restart wazuh-manager
```

Part3: How the Wazuh Dashboard Looks After Integrating ModSecurity & Fail2Ban

1. Dashboard Overview

When you log into the Wazuh Dashboard (<http://<server-ip>:5601/>), the main screen gives you a summary view of your monitored environment.

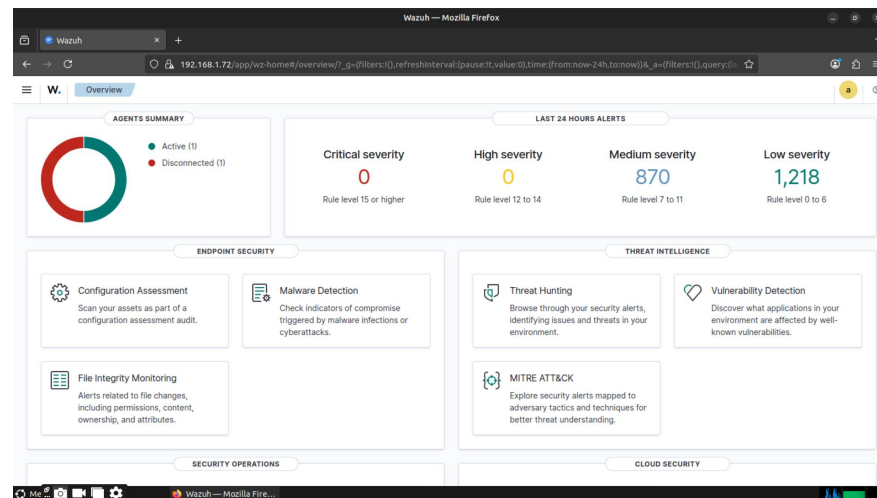
You will see panels and widgets showing:

Total number of agents connected (including your Kali Linux agent).

Number of alerts generated, broken down by severity (Low, Medium, High, Critical).

Top alert categories (e.g., ModSecurity, Fail2Ban, system logs).

Active security events timeline.



2. Agent summary and Log Monitoring

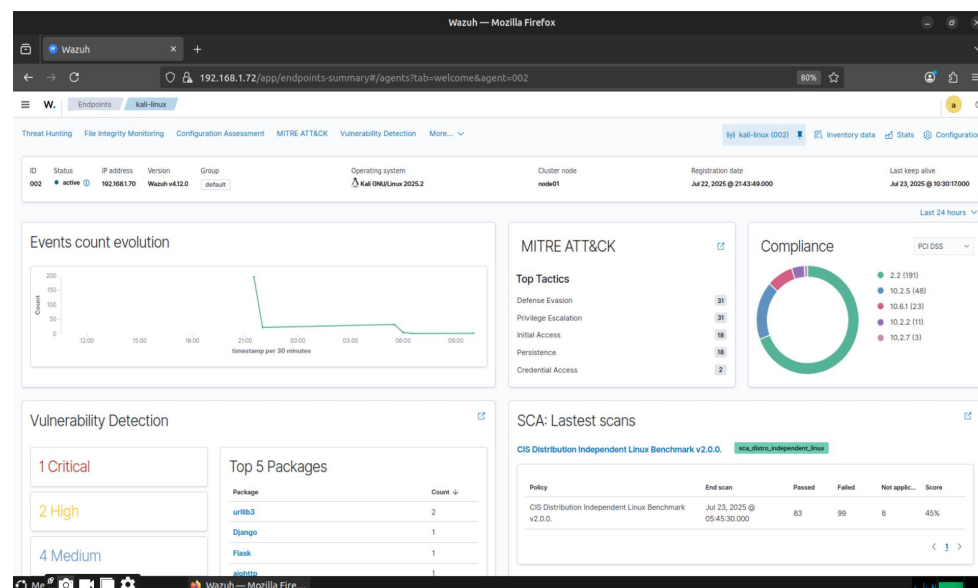
The Agents tab shows all registered agents and their health status.

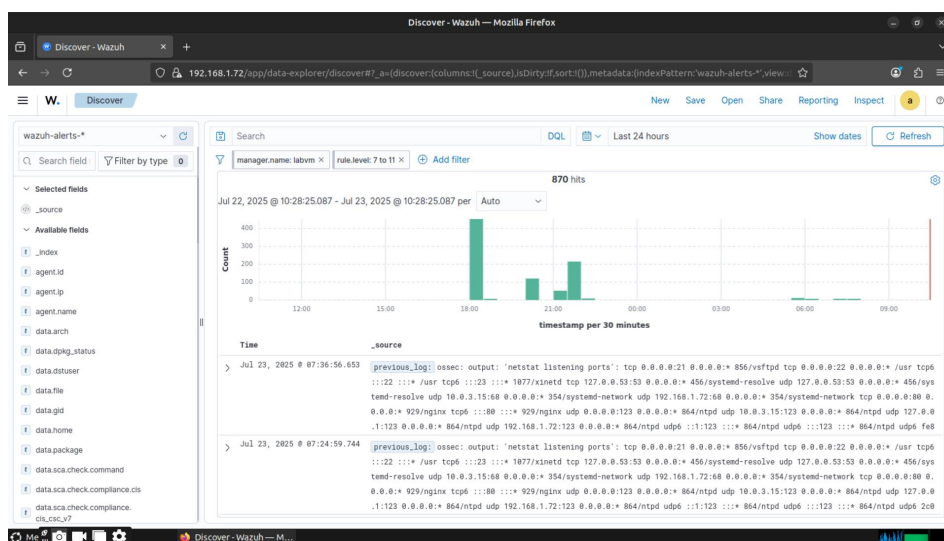
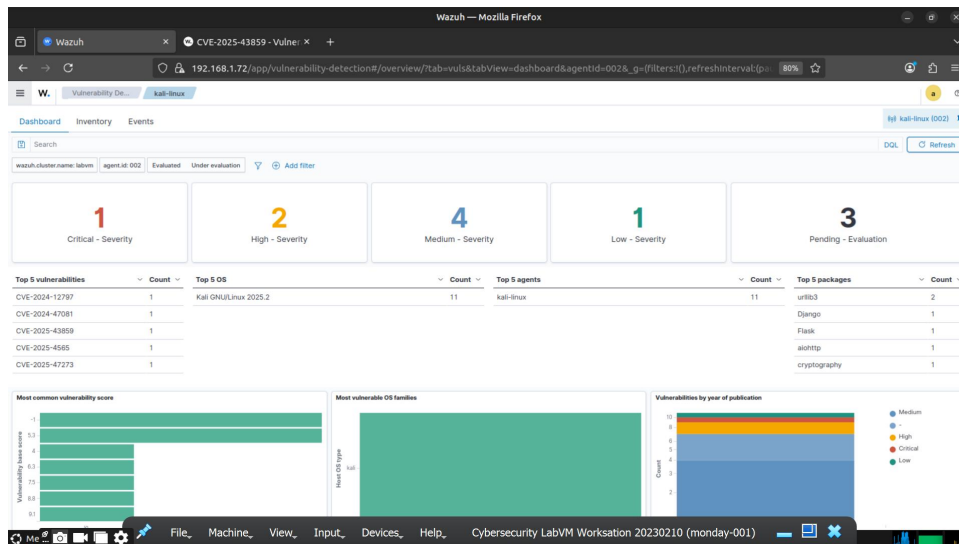
For the Kali Linux agent running ModSecurity and Fail2Ban, you will see:

Last check-in time.

Status (online/offline).

Number of events sent.





Key achievements:

Conclusion: Comprehensive SIEM Integration Using Wazuh, ModSecurity & Fail2Ban

This project successfully demonstrates the end-to-end deployment of a centralized Security Information and Event Management (SIEM) system using Wazuh, OpenSearch, Filebeat, ModSecurity, and Fail2Ban. Through meticulous configuration across multiple platforms, a real-time, scalable, and powerful security monitoring solution was achieved.

Centralized Security Monitoring

Wazuh Manager aggregates security events from agents deployed on remote systems (e.g., Kali Linux) and correlates data from tools like ModSecurity and Fail2Ban for threat detection and incident response.

Log Collection and Visualization

Using Filebeat, logs from the Wazuh Manager are shipped to OpenSearch, enabling structured storage, search, and indexing. The Wazuh Dashboard provides a clear, visual representation of security alerts, agent status, and threat severity.

Agent Integration and Custom Rule Development

The Wazuh Agent was configured on Kali Linux to capture detailed logs from ModSecurity (web application firewall) and Fail2Ban (intrusion prevention). Custom decoders and rules were added to the Wazuh Manager to interpret and act on these logs effectively.

Real-Time Threat Detection

Once integrated, the dashboard displayed alerts in real-time for brute-force attempts, malicious HTTP requests, and other suspicious activities — proving the system's operational success and practical security value.

Scalability & Extensibility

The modular nature of Wazuh allows additional agents and tools to be integrated seamlessly in the future, such as Suricata, ClamAV, or OSQuery, making it a future-proof SIEM foundation.

Security Impact:

Security events from multiple sources are collected in real time.

Alerts are automatically triggered based on pre-defined or custom rules.

Centralized visibility enables quicker incident response and improved security posture.

Administrators gain full transparency into attack patterns across their infrastructure.