wazuh. TRAINING

Module

Edition 2025

Arranged by :

Muhd Akmalul Hakim

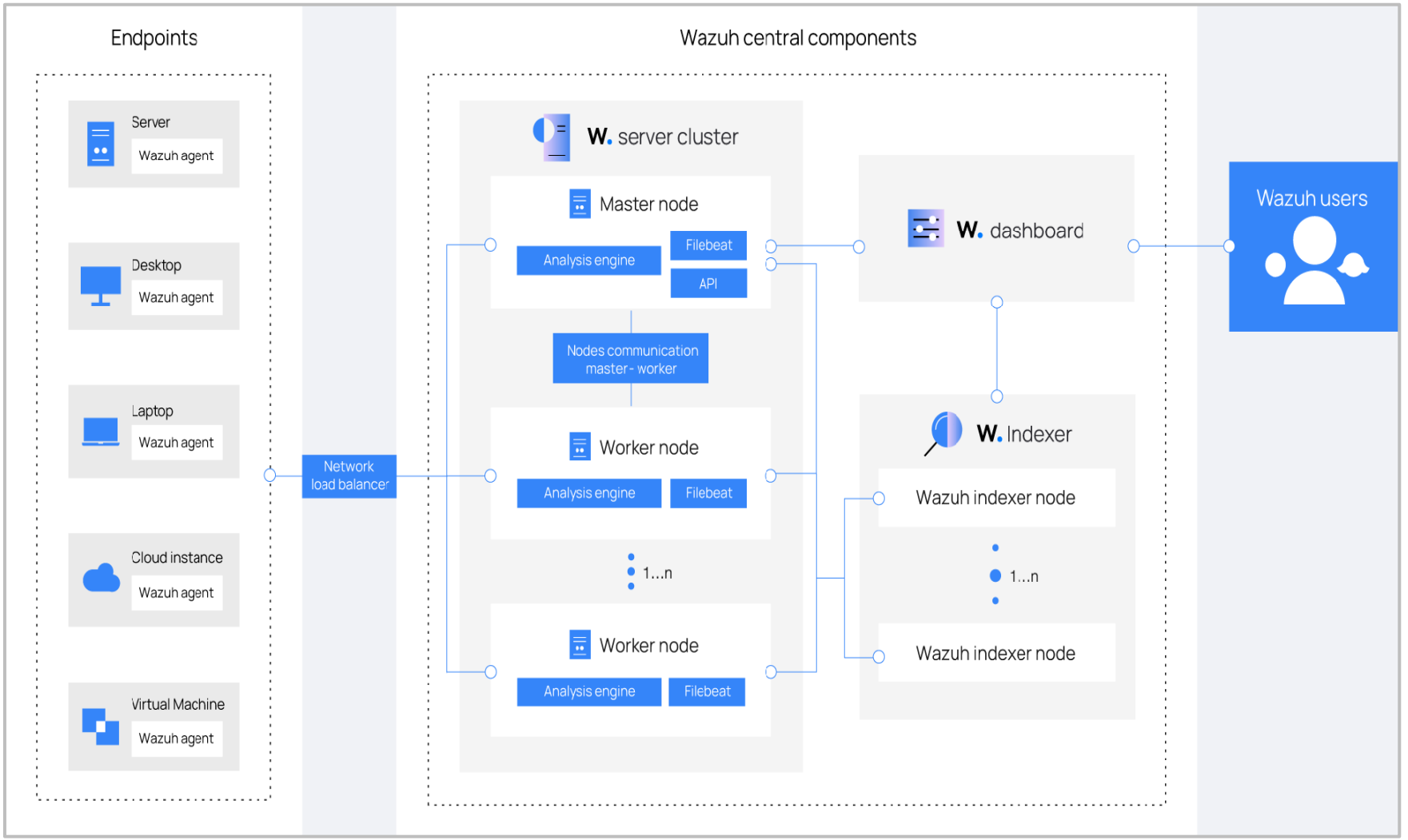
www.orengacademy.com

CONTENT

Edition 2025

WAZUH TRAINING MODULE

Intro: Architecture Wazuh Deployment



Intro: Requirements for Wazuh

Hardware – all in one

The minimum requirements for 25 agents and 90 days of history are as follows:

- 4 CPU
- GB RAM
- 50 GB available disk space – preferably SSD

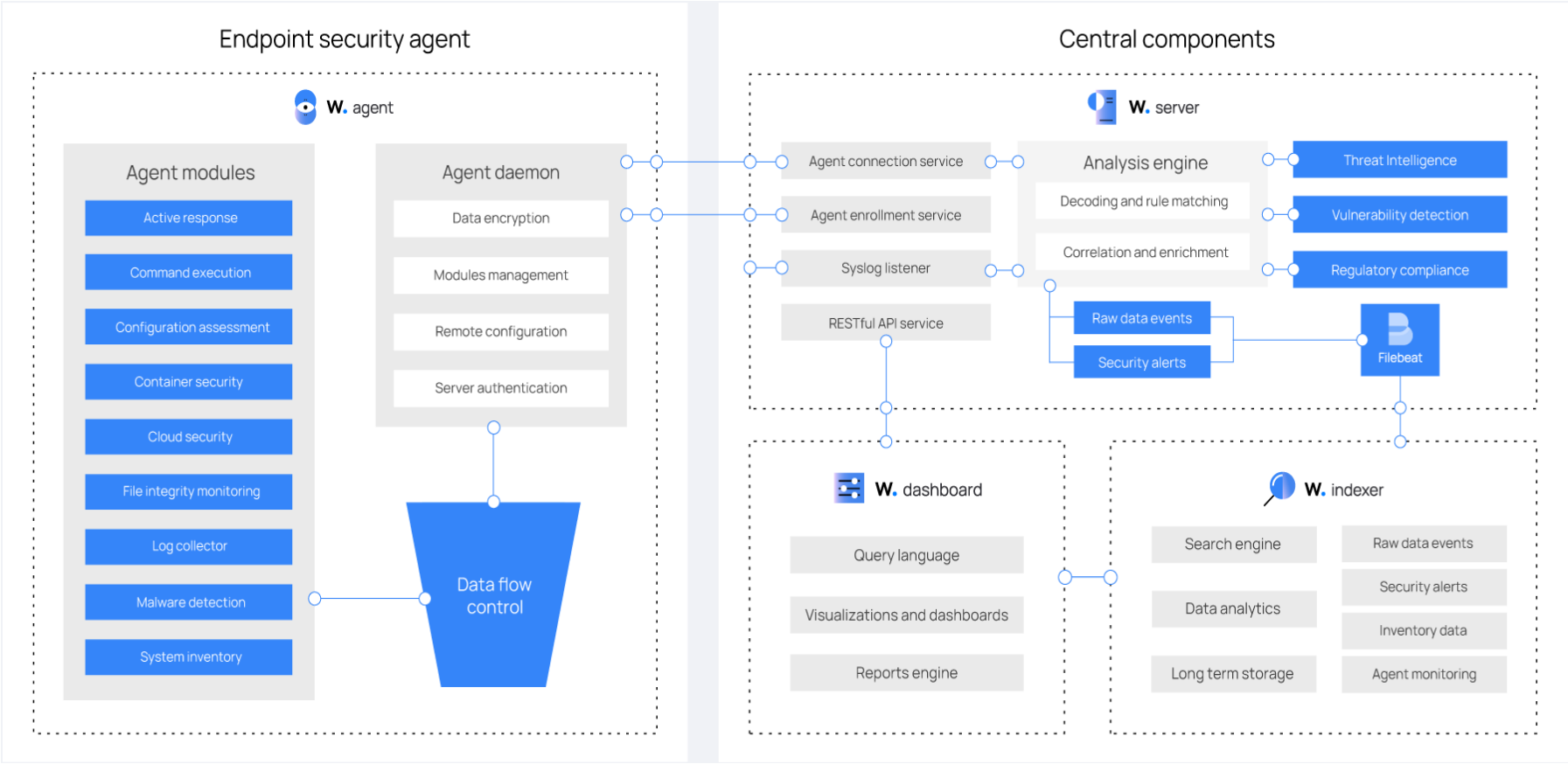
Recommended operating systems

- CentOS 7, 8
- Ubuntu 16.04, 18.04, 20.04, 22.04
- Red Hat Enterprise Linux 7, 8, 9
- Amazon Linux 2

WAZUH TRAINING MODULE

Intro: Component in Wazuh

- Wazuh has four components which is Wazuh Indexer, Wazuh Server, Wazuh Dashboard & Wazuh Agent.



Wazuh Indexer

- Wazuh Indexer

🔍 **Key Functions of Wazuh Indexer:**

1. **Indexing Security Data**:
   It receives processed event data from the Wazuh manager (e.g., log alerts, system events, FIM results) and stores them in a searchable format.
2. **Powering Searches and Visualizations**:
   The data in the Indexer supports real-time queries through the Wazuh Dashboard, enabling you to:
   - Search logs and alerts
   - Create visualizations (e.g., graphs, timelines)
   - Build custom dashboards for security monitoring
3. **Data Retention and Performance**:
   It handles how long data is stored, how it is organized (by indices), and ensures performance for large-scale environments.

Installation Wazuh Indexer

- Certificates Creation

Generating SSL certificates

1. Download the wazuh-certs-tool.sh script and the config.yml configuration file. This creates certificates that encrypt communications between the Wazuh central components.

```
# curl -sO https://packages.wazuh.com/4.11/wazuh-certs-tool.sh
#  curl -sO https://packages.wazuh.com/4.11/config.yml
```

2. Edit ./config.yml and replace the node names and IP values with the corresponding names and IP addresses. You need to do this for all Wazuh server, Wazuh indexer, and Wazuh dashboard nodes. Add as many node fields as needed.

```
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: "<indexer-node-ip>"
    #- 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: "<wazuh-manager-ip>"
    #  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
  dashboard:
    - name: dashboard
      ip: "<dashboard-node-ip>"
```

3. Run ./wazuh-certs-tool.sh to create the certificates. For a multi-node cluster, these certificates need to be later deployed to all Wazuh instances in your cluster.

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

4. Compress all the necessary files

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

5. Copy the wazuh-certificates.tar file to all the nodes, including the Wazuh indexer, Wazuh server, and Wazuh dashboard nodes. This can be done by using the scp utility.

Nodes Installation

Installing package dependencies

1. Install the following packages if missing:

```
# apt-get install debconf adduser procps
```

Adding the Wazuh repository

1. Install the following packages if missing.

```
# apt-get install gnupg apt-transport-https
```

2. 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
```

3. 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
```

4. Update the packages information.

```
# apt-get update
```

Installing the Wazuh indexer

1. Install the Wazuh indexer package.

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

Configuring the Wazuh indexer

1. Edit the /etc/wazuh-indexer/opensearch.yml configuration file and replace the following values:
   a) **network.host**: Sets the address of this node for both HTTP and transport traffic. The node will bind to this address and use it as its publish address. Accepts an IP address or a hostname.
   Use the same node address set in config.yml to create the SSL certificates.
   b) **node.name**: Name of the Wazuh indexer node as defined in the config.yml file. For example, node-1.
   c) cluster.initial_master_nodes: List of the names of the master-eligible nodes. These names are defined in the config.yml file. Uncomment the node-2 and node-3 lines, change the names, or add more lines, according to your config.yml definitions**.**

```
cluster.initial_master_nodes:
- "node-1"
- "node-2"
- "node-3"
```

d) **discovery.seed_hosts:** List of the addresses of the master-eligible nodes. Each element can be either an IP address or a hostname. You may leave this setting commented if you are configuring the Wazuh indexer as a single node. For multi-node configurations, uncomment this setting and set the IP addresses of each master-eligible node.

```
discovery.seed_hosts:
 - "10.0.0.1"
 - "10.0.0.2"
 - "10.0.0.3"
```

e) **plugins.security.nodes_dn:** List of the Distinguished Names of the certificates of all the Wazuh indexer cluster nodes. Uncomment the lines for node-2 and node-3 and change the common names (CN) and values according to your settings and your config.yml definitions.

Deploying certificates

Note: Make sure that a copy of the Wazuh certificates.tar file, created during the initial configuration step, is placed in your working directory.

1. Run the following commands replacing <INDEXER_NODE_NAME> with the name of the Wazuh indexer node you are configuring as defined in config.yml. For example, node-1. This deploys the SSL certificates to encrypt communications between the Wazuh central components.

```
# NODE_NAME=<INDEXER_NODE_NAME>
# mkdir /etc/wazuh-indexer/certs
# tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/
./$NODE_NAME.pem ./$NODE_NAME-key.pem ./admin.pem
./admin-key.pem ./root-ca.pem
# mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem
/etc/wazuh-indexer/certs/indexer.pem
# mv -n /etc/wazuh-indexer/certs/$NODE_NAME-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
```

2. **Recommended action**: If no other Wazuh components are going to be installed on this node, remove the wazuh-certificates.tar file by running rm -f ./wazuh-certificates.tar to increase security.

Starting the service

1. Enable and start the Wazuh indexer service.

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

Repeat this stage of the installation process for every Wazuh indexer node in your cluster. Then proceed with initializing your single-node or multi-node cluster in the next stage.

Cluster initialization

1. Run the Wazuh indexer indexer-security-init.sh script on *any* Wazuh indexer node to load the new certificates information and start the single-node or multi-node cluster.

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

Note: You only have to initialize the cluster *once*, there is no need to run this command on every node.

Testing the cluster installation

1. Replace <WAZUH_INDEXER_IP_ADDRESS> and run the following commands to confirm that the installation is successful.

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

2. Replace <WAZUH_INDEXER_IP_ADDRESS> and run the following command to check if the single-node or multi-node cluster is working correctly.

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

Wazuh Server

- Wazuh Server

Main Functions of the Wazuh Server:

1. **Log Collection & Aggregation**
    - Receives security event data, logs, and telemetry from **Wazuh Agents** installed on endpoints (Windows, Linux, macOS).
2. **Log Analysis & Correlation**
    - Parses and analyzes logs using **decoders** and **rules**.
    - Detects threats such as brute-force attempts, malware infections, misconfigurations, and unauthorized activity.
3. **Alert Generation**
    - When suspicious or malicious behavior is detected, it creates alerts (stored in /var/ossec/logs/alerts/alerts.json).
    - Alerts are forwarded to the **Wazuh Indexer** for searching and visualization.
4. **Configuration Management**

- Sends configuration files to agents (e.g., rules, syscheck, rootcheck).
- Centralized management allows you to easily push updates and monitor agent status.

5. **Security Modules Execution**
   - File Integrity Monitoring (FIM)
   - Rootkit Detection
   - Vulnerability Detection (via integration with vulnerability feeds)
   - Log Collection & Forwarding (Syslog, journald, etc.)

Installation Wazuh Server

Wazuh server node installation

1. Install the following packages if missing.

```
# apt-get install gnupg apt-transport-https
```

2. 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
```

3. 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
```

4. Update the packages information.

```
# apt-get update
```

Installing the Wazuh manager

1. Install the Wazuh manager package.

```
# apt-get -y install wazuh-manager
```

Installing Filebeat

1. Install the Filebeat package.

```
# curl -O
https://packages.wazuh.com/4.x/apt/pool/main/f/filebeat/
filebeat-oss-7.10.2-amd64.deb

# apt-get install ./filebeat-oss-7.10.2-amd64.deb
```

Configuring Filebeat

1. Download the preconfigured Filebeat configuration file.

```
# curl -so /etc/filebeat/filebeat.yml
https://packages.wazuh.com/4.11/tpl/wazuh/filebeat/filebeat.yml
```

2. Edit the /etc/filebeat/filebeat.yml configuration file and replace the following value:
   a) **hosts**: The list of Wazuh indexer nodes to connect to. You can use either IP addresses or hostnames. By default, the host is set to localhost hosts: ["127.0.0.1:9200"]. Replace it with

your Wazuh indexer address accordingly. If you have more than one Wazuh indexer node, you can separate the addresses using commas. For example, hosts: ["10.0.0.1:9200", "10.0.0.2:9200", " 10.0.0.3:9200"]

```
# Wazuh - Filebeat configuration file

output.elasticsearch:

        hosts: ["10.0.0.1:9200"]
        protocol: https
        username: ${username}
        password: ${password}
```

3. Create a Filebeat keystore to securely store authentication credentials.

```
# filebeat keystore create
```

4. Add the default username and password admin:admin to the secrets keystore.

```
# echo admin | filebeat keystore add username --stdin --force

# echo admin | filebeat keystore add password --stdin –force
```

5. Download the alerts template for the Wazuh indexer.

```
# curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/v4.11.2/extensions/elasticsearch/7.x/wazuh-template.json

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

6. Install the Wazuh module for Filebeat.

```
# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.4.tar.gz | tar -xvz -C /usr/share/filebeat/module
```

Deploying certificates

Note: Make sure that a copy of the wazuh-certificates.tar file, created during the initial configuration step, is placed in your working directory.

1. Replace <SERVER_NODE_NAME> with your Wazuh server node certificate name, the same one used in config.yml when creating the certificates. Then, move the certificates to their corresponding location.

```
# NODE_NAME=<SERVER_NODE_NAME>

# mkdir /etc/filebeat/certs

# tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem

# mv -n /etc/filebeat/certs/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem

# mv -n /etc/filebeat/certs/$NODE_NAME-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
```

Configuring the Wazuh indexer connection

1. Save the Wazuh indexer username and password into the Wazuh manager keystore using the wazuh-keystore tool:

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

2. Edit /var/ossec/etc/ossec.conf to configure the indexer connection. By default, the indexer settings have one host configured. It's set to 0.0.0.0 as highlighted below.

```
<indexer>
 <enabled>yes</enabled>
 <hosts>
  <host>https://0.0.0.0:9200</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>
```

3. Replace 0.0.0.0 with your Wazuh indexer node IP address or hostname. You can find this value in the Filebeat config file /etc/filebeat/filebeat.yml. Ensure the Filebeat certificate and key name match the certificate files in /etc/filebeat/certs. If you have a Wazuh indexer cluster, add a <host> entry for each one of your nodes. For example, in a two-nodes configuration:

```
<hosts>
    <host>https://10.0.0.1:9200</host>
    <host>https://10.0.0.2:9200</host>
</hosts>
```

Starting the Wazuh manager

1. Enable and start the Wazuh manager service.

```
# systemctl daemon-reload
# systemctl enable wazuh-manager
# systemctl start wazuh-manager
```

2. Run the following command to verify the Wazuh manager status.

```
# systemctl status wazuh-manager
```

Starting the Filebeat service

1. Enable and start the Filebeat service.

```
# systemctl daemon-reload

# systemctl enable filebeat

# systemctl start filebeat
```

Wazuh Dashboard

- Wazuh Dashboard

🖥️ **Main Functions of the Wazuh Dashboard:**

1. **Real-time Security Monitoring**
    - View alerts, logs, and system events in real time.
    - Dashboards are customizable with graphs, tables, and timelines.
2. **Log Search and Filtering**
    - Query indexed data from the Wazuh Indexer using advanced filters.
    - Drill down into specific events or indicators of compromise.

3. **Visual Analytics**
    - Prebuilt dashboards for:
        - File Integrity Monitoring (FIM)
        - Rootkit detection
        - Vulnerability detection
        - Authentication attempts
        - Malware alerts
4. **Agent Management Interface**
    - View status of all registered agents (active, disconnected, etc.)
    - Group agents, check logs, and restart agent services.
5. **Rule & Decoder Visualization**
    - Understand which rules triggered alerts.
    - View rule IDs, severity levels, and categories.
6. **Security Operations Center (SOC) View**
    - Centralizes all alerting, enabling analysts to monitor the entire infrastructure in one place.

Installation Wazuh Dashboard

Wazuh dashboard installation

Installing package dependencies

1. Install the following packages if missing.

```
# apt-get install debhelper tar curl libcap2-bin #debhelper version 9 or later
```

Adding the Wazuh repository

1. Install the following packages if missing.

```
# apt-get install gnupg apt-transport-https
```

2. 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
```

3. 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
```

4. Update the packages information.

```
# apt-get update
```

Installing the Wazuh dashboard

1. Install the Wazuh dashboard package.

```
# apt-get -y install wazuh-dashboard
```

Configuring the Wazuh dashboard

1. Edit the /etc/wazuh-dashboard/opensearch_dashboards.yml file and replace the following values:
   a) **server.host**: This setting specifies the host of the Wazuh dashboard server. To allow remote users to connect, set the value to the IP address or DNS name of the Wazuh dashboard server. The value 0.0.0.0 will accept all the available IP addresses of the host.
   b) **opensearch.hosts:** The URLs of the Wazuh indexer instances to use for all your queries. The Wazuh dashboard can be configured to connect to multiple Wazuh indexer nodes in the same cluster. The addresses of the nodes can be separated by commas. For example, ["https://10.0.0.2:9200", "https://10.0.0.3:9200","https://10.0.0.4:9200"]

```
server.host: 0.0.0.0

server.port: 443

opensearch.hosts: https://localhost:9200

opensearch.ssl.verificationMode: certificate
```

Deploying certificates

Note: Make sure that a copy of the wazuh-certificates.tar file, created during the initial configuration step, is placed in your working directory.

1. Replace <DASHBOARD_NODE_NAME> with your Wazuh dashboard node name, the same one used in config.yml to create the certificates and move the certificates to their corresponding location.

```
# NODE_NAME=<DASHBOARD_NODE_NAME>

# mkdir /etc/wazuh-dashboard/certs

# tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem

# mv -n /etc/wazuh-dashboard/certs/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem

# mv -n /etc/wazuh-dashboard/certs/$NODE_NAME-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
```

Starting the Wazuh dashboard service

1. Enable and start the Wazuh dashboard service.

```
# systemctl daemon-reload

# systemctl enable wazuh-dashboard

# systemctl start wazuh-dashboard
```

2. Edit the /usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml file and replace the url value with the IP address or hostname of the Wazuh server master node.

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

Wazuh Agent

- The **Wazuh Agent** is a lightweight software installed on endpoints (like servers, desktops, or cloud instances) that **collects security-relevant data** and sends it to the **Wazuh Server** for analysis.

🧱 Main Functions of the Wazuh Agent:

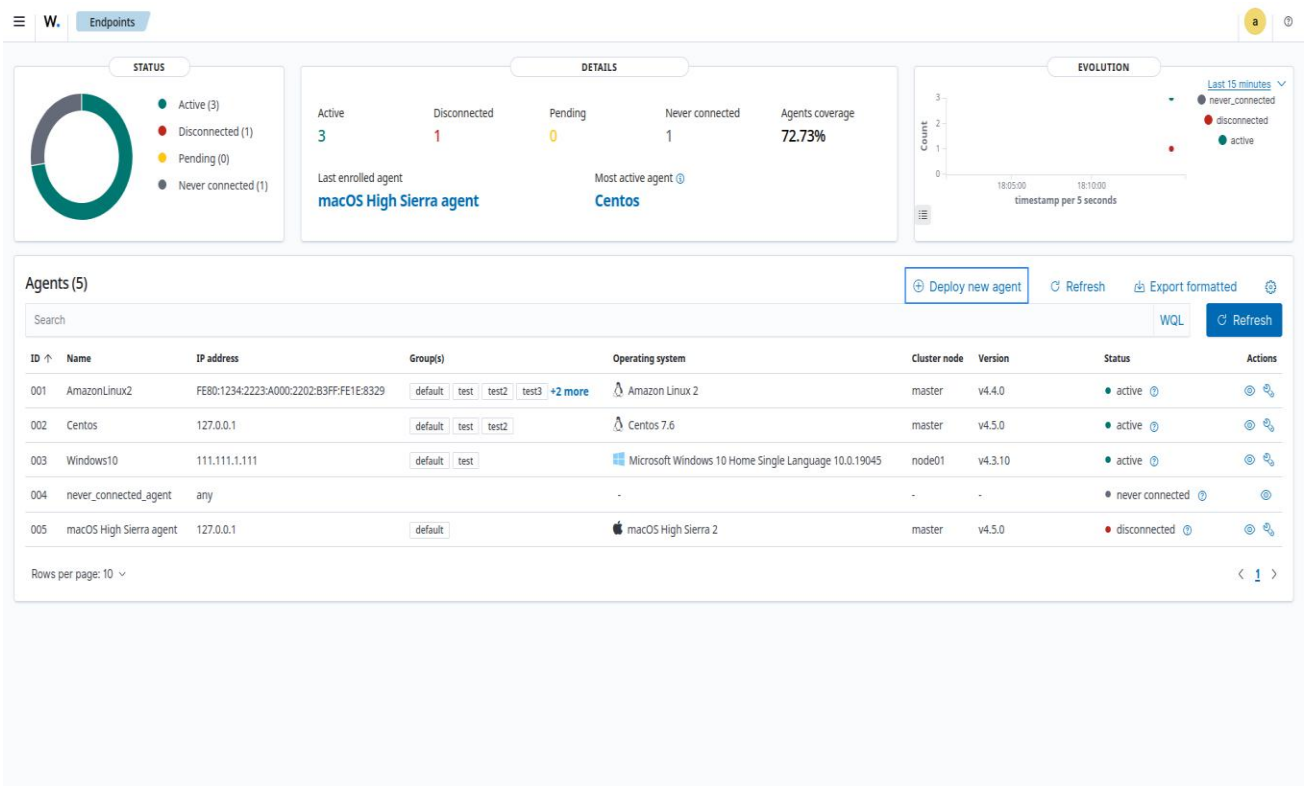
1. **Log Collection**
    - Collects logs from:
        - System logs (e.g., /var/log/, Windows Event Logs)
        - Application logs
        - Cloud services (if configured)
    - Forwards them securely to the Wazuh Server.
2. **File Integrity Monitoring (FIM)**
    - Detects changes to critical files (e.g., configuration files, binaries).
    - Alerts on modifications, creations, or deletions.
3. **Rootkit and Malware Detection**
    - Scans for rootkits, suspicious processes, hidden ports, and anomalous behavior.

4. **Configuration Assessment**
    - Evaluates system configuration against security policies (e.g., CIS Benchmarks).
    - Helps in compliance audits.
5. **Active Response**
    - Can execute automated response scripts on the endpoint (e.g., block IP, restart service, isolate host).
6. **Vulnerability Detection**
    - Collects package info and sends it to the Wazuh Server for matching against known vulnerabilities.
7. **Real-Time Monitoring**
    - Monitors login attempts, privilege escalations, and access to sensitive areas.

Installation Wazuh Agent

To install a Wazuh agent, select your operating system and follow the instructions.

You can also deploy a new agent following the instructions in the Wazuh dashboard. Go to Agents management > Summary, and click on Deploy new agent.



Then the Wazuh dashboard will show you the steps to deploy a new agent.

Close

Deploy new agent

Select the package to download and install on your system:

LINUX	WINDOWS	macOS
RPM amd64	MSI 32/64 bits	Intel
RPM aarch64		Apple silicon
DEB amd64		
DEB aarch64		

For additional systems and architectures, please check our **documentation**.

Server address:

This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FQDN).

Assign a server address

10.0.0.198

Remember server address

Optional settings:

By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name in the field below.

Assign an agent name:

Windows

The agent name must be unique. It can't be changed once the agent has been enrolled.

Select one or more existing groups:

default

4 **Run the following commands to download and install the agent:**

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.8.0-1.msi -OutFile $
{env.tmp}\wazuh-agent; msiexec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='10.0.0.198'
WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='Windows'
```

Requirements

- You will need administrator privileges to perform this installation.
- PowerShell 3.0 or greater is required.

Keep in mind you need to run this command in a Windows PowerShell terminal.

5 **Start the agent:**

```
NET START WazuhSvc
```

Close

Proof of Concept Guide

File integrity monitoring

File Integrity Monitoring (FIM) helps in auditing sensitive files and meeting regulatory compliance requirements. Wazuh has an inbuilt FIM module that monitors file system changes to detect the creation, modification, and deletion of files.

This use case uses the Wazuh FIM module to detect changes in monitored directories on Ubuntu and Windows endpoints. The Wazuh FIM module enriches alert data by fetching information about the user and process that made the changes using who-data audit.

Infrastructure

Endpoint	Description
Ubuntu 22.04	The Wazuh FIM module monitors a directory on this endpoint to detect file creation, changes, and deletion.
Windows 11	The Wazuh FIM module monitors a directory on this endpoint to detect file creation, changes, and deletion.

Configuration

Ubuntu endpoint

Perform the following steps to configure the Wazuh agent to monitor filesystem changes in the /root directory.

1. Edit the Wazuh agent /var/ossec/etc/ossec.conf configuration file. Add the directories for monitoring within the <syscheck> block. For this use case, you configure Wazuh to monitor the /root directory. To get additional information about the user and process that made the changes, enable who-data audit:

```
<directories check_all="yes" report_changes="yes" realtime="yes">/root</directories>
```

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

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

Windows endpoint

Take the following steps to configure the Wazuh agent to monitor filesystem changes in the**:**

```
C:\Users\Administrator\Desktop
```

1. Edit the C:\Program Files (x86)\ossec-agent\ossec.conf configuration file on the monitored Windows endpoint. Add the directories for monitoring within the <syscheck> block. For this use case, you configure Wazuh to monitor the C:\Users\Administrator\Desktop directory. To get additional information about the user and process that made the changes, enable who-data audit:

```
<directories check_all="yes" report_changes="yes" realtime="yes">C:\Users\<USER_NAME>\Desktop</directories>
```

2. Restart the Wazuh agent using Powershell with administrator privileges to apply the changes:

```
Restart-Service -Name Wazuh
```

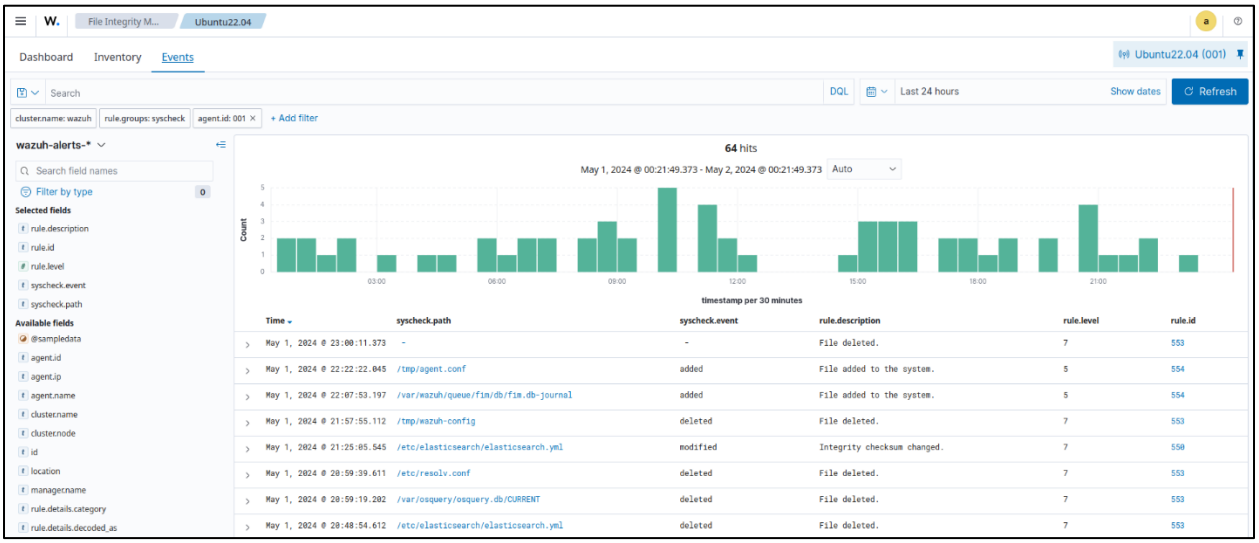
Test the configuration

1. Create a text file in the monitored directory then wait for 5 seconds.
2. Add content to the text file and save it. Wait for 5 seconds.
3. Delete the text file from the monitored directory.

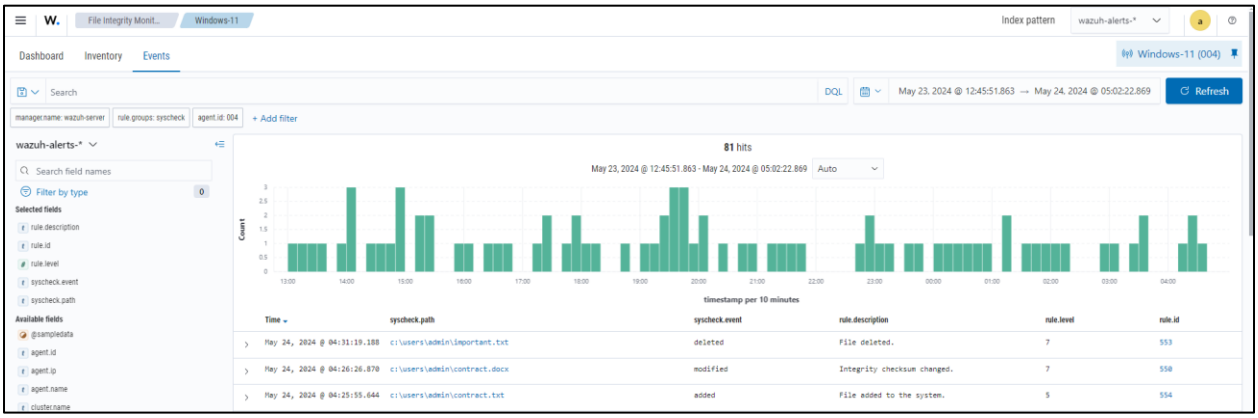
Visualize the alerts

You can visualize the alert data in the Wazuh dashboard. To do this, go to the File Integrity Monitoring module and add the filters in the search bar to query the alerts:

Ubuntu - rule.id: is one of 550,553,554



Windows - rule.id: is one of 550,553,554



Detecting a brute-force attack

Brute-forcing is a common attack vector that threat actors use to gain unauthorized access to endpoints and services. Services like SSH on Linux endpoints and RDP on Windows endpoints are usually prone to brute-force attacks. Wazuh identifies brute-force attacks by correlating multiple authentication failure events.

The section on Blocking attacks with Active Response describes how to configure an active response to block the IP address of an attacker. In this use case, we show how Wazuh detects brute-force attacks on RHEL and Windows endpoints.

Infrastructure

Endpoint	Description
Ubuntu 22.04	Attacker endpoint that performs brute-force attacks. It's required to have an SSH client installed on this endpoint.
RHEL 9.0	Victim endpoint of SSH brute-force attacks. It's required to have an SSH server installed and enabled on this endpoint.
Windows 11	Victim endpoint of RDP brute-force attacks. It's required to enable RDP on this endpoint.

Configuration

Perform the following steps to configure the Ubuntu endpoint. This allows performing authentication failure attempts on the monitored RHEL and Windows endpoints.

1. On the attacker endpoint, install Hydra and use it to execute the brute-force attack:

```
sudo apt update
sudo apt install -y hydra
```

Attack emulation

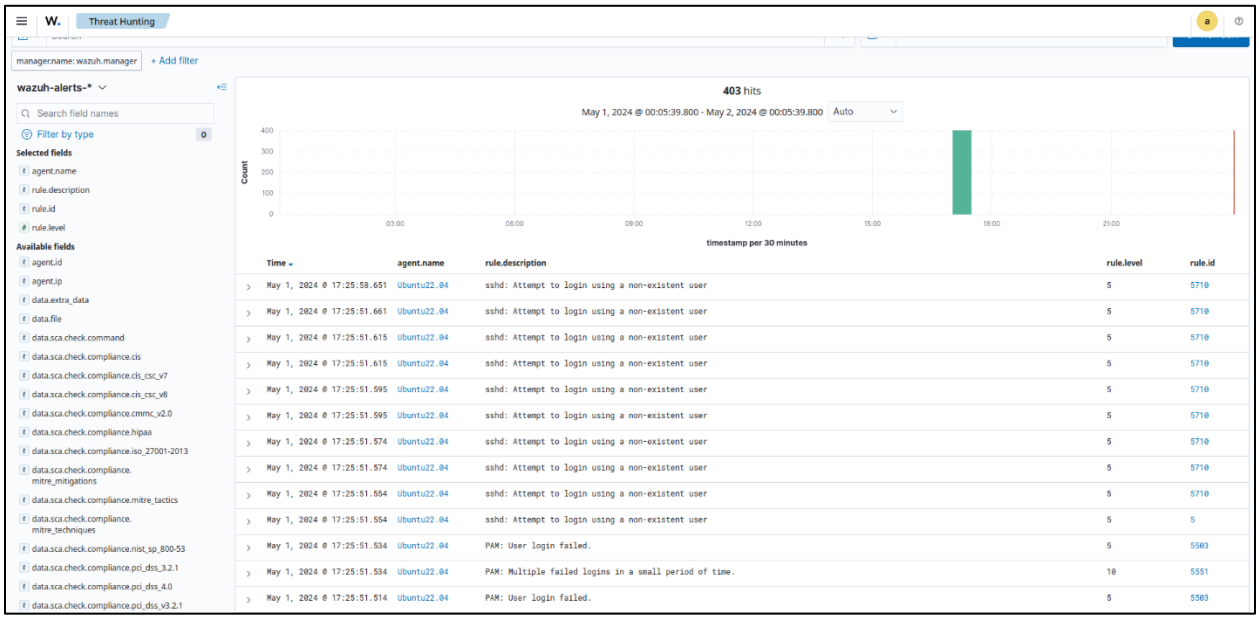
1. Create a text file with 10 random passwords.
2. Run Hydra from the attacker endpoint to execute brute-force attacks against the RHEL endpoint. To do this, replace <RHEL_IP> with the IP address of the RHEL endpoint and run the command below:

```
sudo hydra -L <USRNAME_LIST.txt> -P <PASSWD_LIST.txt> <RHEL_IP> ssh
```

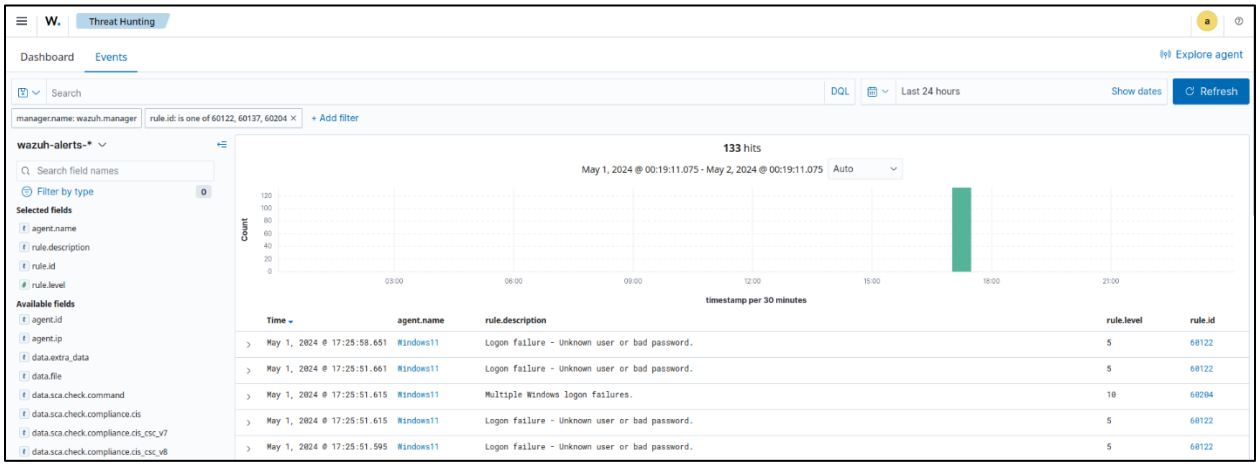
Visualize the alerts

You can visualize the alert data in the Wazuh dashboard. To do this, go to the Threat Hunting module and add the filters in the search bar to query the alerts.

- Linux - rule.id:(5551 OR 5712). Other related rules are 5710, 5711, 5716, 5720, 5503, 5504.



- Windows - rule.id:(60122 OR 60204)



Detecting an SQL injection attack

You can use Wazuh to detect SQL injection attacks from web server logs that contain patterns like select, union, and other common SQL injection patterns.

SQL injection is an attack in which a threat actor inserts malicious code into strings transmitted to a database server for parsing and execution. A successful SQL injection attack gives unauthorized access to confidential information contained in the database.

In this use case, you simulate an SQL injection attack against an Ubuntu endpoint and detect it with Wazuh.

Infrastructure

Endpoint	Description
Ubuntu 22.04	Victim endpoint running an Apache 2.4.54 web server.
RHEL 9.0	Attacker endpoint that launches the SQL injection attack.

Configuration

Ubuntu endpoint

Perform the following steps to install Apache and configure the Wazuh agent to monitor the Apache logs.

1. Update the local packages and install the Apache web server:

```
sudo apt update
sudo apt install apache2
```

2. If the firewall is enabled, modify it to allow external access to web ports. Skip this step if the firewall is disabled.

```
sudo ufw app list
sudo ufw allow 'Apache'
sudo ufw status
```

3. Check the status of the Apache service to verify that the web server is running:

```
sudo systemctl status apache2
```

4. Use the curl command or open http://<UBUNTU_IP> in a browser to view the Apache landing page and verify the installation:

```
curl http://<UBUNTU_IP>
```

5. Add the following lines to the Wazuh agent /var/ossec/etc/ossec.conf file. This allows the Wazuh agent to monitor the access logs of your Apache server:

```
<ossec_config>
 <localfile>
  <log_format>apache</log_format>
  <location>/var/log/apache2/access.log</location>
 </localfile>
</ossec_config>
```

6. Restart the Wazuh agent to apply the configuration changes:

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

Attack emulation

Replace <UBUNTU_IP> with the appropriate IP address and execute the following command from the attacker endpoint:

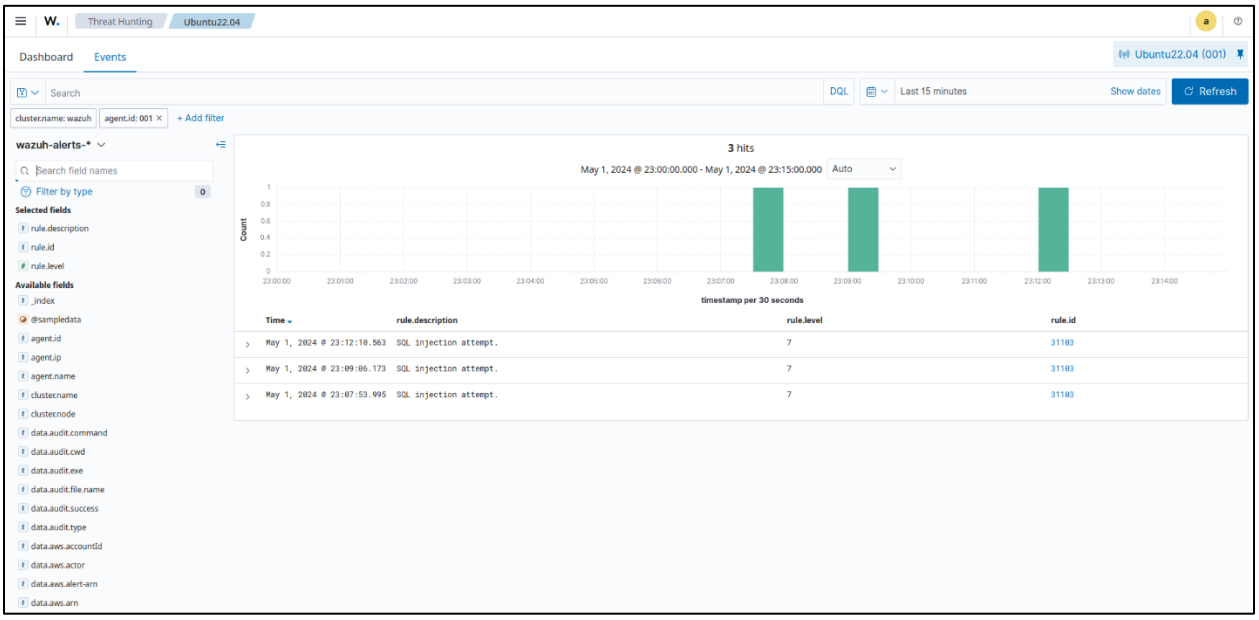
```
curl -XGET
"http://<UBUNTU_IP>/users/?id=SELECT+*+FROM+users";
```

The expected result here is an alert with rule ID 31103 but a successful SQL injection attempt generates an alert with rule ID 31106.

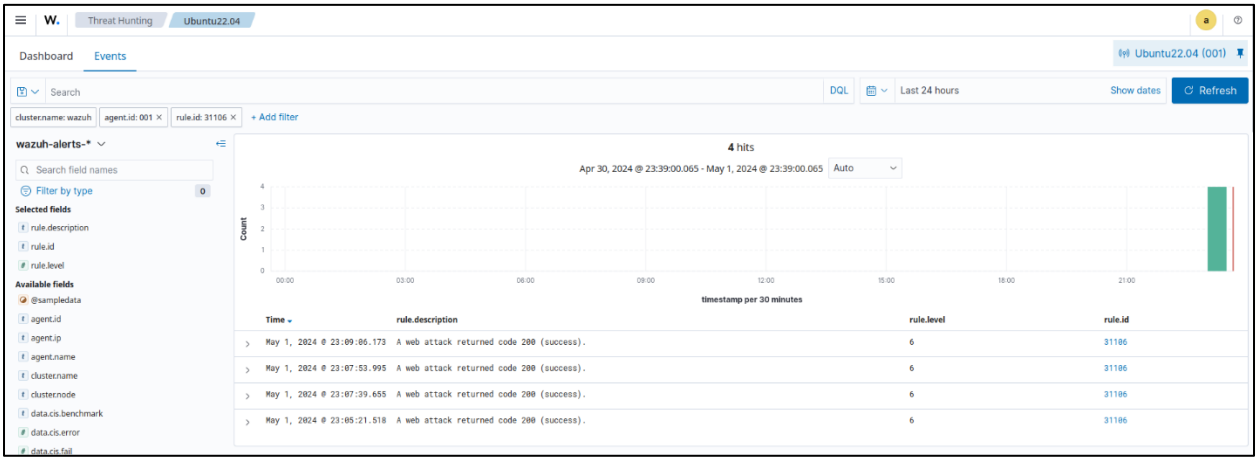
Visualize the alerts

You can visualize the alert data in the Wazuh dashboard. To do this, go to the Threat Hunting module and add the filters in the search bar to query the alerts.

- rule.id:31103



- rule.id:31106



Detecting malware using YARA integration

You can use the YARA integration with Wazuh to scan files added or modified on an endpoint for malware. YARA is a tool to detect and classify malware artifacts.

In this use case, we demonstrate how to configure YARA with Wazuh to detect malware on Linux and Windows endpoints.

Infrastructure

Endpoint	Description
Ubuntu 22.04 / RHEL 9.0	The YARA Active Response module scans new or modified files whenever the Wazuh FIM module triggers an alert.
Windows 11	The YARA Active Response module scans new or modified files whenever the Wazuh FIM module triggers an alert.

Configuration

Linux endpoint

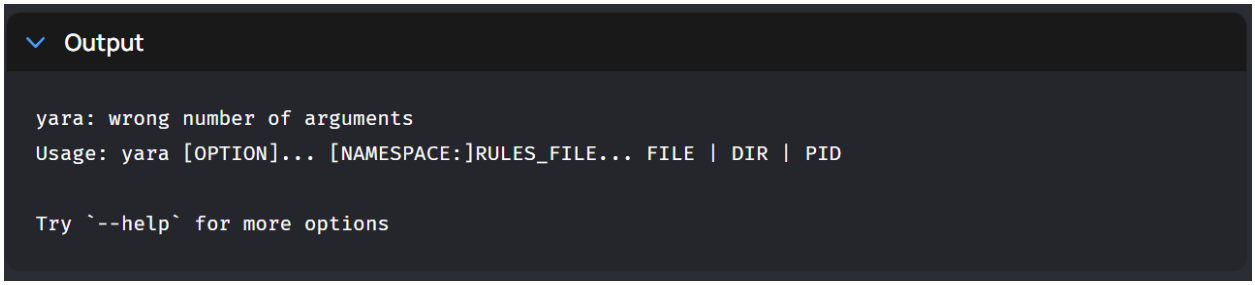
Perform the following steps to install YARA, and configure the Active Response and FIM modules.

1. Download, compile and install YARA:

```
# sudo apt update
# sudo apt install -y make gcc autoconf libtool libssl-dev pkg-config jq
# sudo curl -LO https://github.com/VirusTotal/yara/archive/v4.2.3.tar.gz
# sudo tar -xvzf v4.2.3.tar.gz -C /usr/local/bin/ && rm -f v4.2.3.tar.gz
# cd /usr/local/bin/yara-4.2.3/
# sudo ./bootstrap.sh && sudo ./configure && sudo make && sudo make install && sudo make check
```

2. Test that YARA is running properly:

```
# yara
```



If the error message below is displayed:

```
/usr/local/bin/yara: error while loading shared libraries: libyara.so.9: cannot open shared object file: No such file or directory.
```

This means that the loader doesn't find the libyara library usually located in /usr/local/lib. Add the /usr/local/lib path to the /etc/ld.so.conf loader configuration file to solve this:

```
sudo su

echo "/usr/local/lib" >> /etc/ld.so.conf

ldconfig
```

Switch back to the previous user.

3. Download YARA detection rules:

```
# sudo mkdir -p /tmp/yara/rules
# sudo curl 'https://valhalla.nextron-
systems.com/api/v1/get' \
-H 'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*
/*;q=0.8' \
-H 'Accept-Language: en-US,en;q=0.5' \
--compressed \
-H 'Referer: https://valhalla.nextron-systems.com/' \
-H 'Content-Type: application/x-www-form-urlencoded' \
-H 'DNT: 1' -H 'Connection: keep-alive' -H 'Upgrade-
Insecure-Requests: 1' \
--data
'demo=demo&apikey=1111111111111111111111111111
1111111111111111111111111111111111&format=text' \
-o /tmp/yara/rules/yara_rules.yar
```

4. Create a yara.sh script in the /var/ossec/active-response/bin/ directory. This is necessary for the Wazuh-YARA Active Response scans:

```
#!/bin/bash
# Wazuh - Yara active response
# Copyright (C) 2015-2022, Wazuh Inc.
#
# This program is free software; you can redistribute it
# and/or modify it under the terms of the GNU General Public
# License (version 2) as published by the FSF - Free Software
# Foundation.
#------------------------- Gather parameters -------------------------#
# Extra arguments
read INPUT_JSON
YARA_PATH=$(echo $INPUT_JSON | jq -r .parameters.extra_args[1])
YARA_RULES=$(echo $INPUT_JSON | jq -r .parameters.extra_args[3])
FILENAME=$(echo $INPUT_JSON | jq -r .parameters.alert.syscheck.path)
# Set LOG_FILE path
LOG_FILE="logs/active-responses.log"

size=0
actual_size=$(stat -c %s ${FILENAME})
while [ ${size} -ne ${actual_size} ]; do
  sleep 1
  size=${actual_size}
  actual_size=$(stat -c %s ${FILENAME})
done
#------------------------ Analyze parameters -------------------------#
if [[ ! $YARA_PATH ]] || [[ ! $YARA_RULES ]]
then
  echo "wazuh-yara: ERROR - Yara active response error. Yara path and rules parameters are
mandatory." >> ${LOG_FILE}
  exit 1
fi
#------------------------- Main workflow -------------------------#
# Execute Yara scan on the specified filename
yara_output="$("${YARA_PATH}"/yara -w -r "$YARA_RULES" "$FILENAME")"

if [[ $yara_output != "" ]]
then
  # Iterate every detected rule and append it to the LOG_FILE
  while read -r line; do
    echo "wazuh-yara: INFO - Scan result: $line" >> ${LOG_FILE}
  done <<< "$yara_output"
fi

exit 0;
```

5. Change yara.sh file owner to root:wazuh and file permissions to 0750:

```
# sudo chown root:wazuh /var/ossec/active-response/bin/yara.sh
# sudo chmod 750 /var/ossec/active-response/bin/yara.sh
```

6. Add the following within the <syscheck> block of the Wazuh agent /var/ossec/etc/ossec.conf configuration file to monitor the /tmp/yara/malware directory:

```
<directories realtime="yes">/tmp/yara/malware</directories>
```

7. Restart the Wazuh agent to apply the configuration changes:

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

Wazuh server

Perform the following steps to configure Wazuh to alert for file changes in the endpoint monitored directory. The steps also configure an active response script to trigger whenever a suspicious file is detected.

1. Add the following rules to the /var/ossec/etc/rules/local_rules.xml file. The rules

detect FIM events in the monitored directory. They also alert when the YARA integration finds malware. You can modify the rules to detect events from other directories:

```
<group name="syscheck,">
 <rule id="100300" level="7">
  <if_sid>550</if_sid>
  <field name="file">/tmp/yara/malware/</field>
  <description>File modified in /tmp/yara/malware/
directory.</description>
 </rule>
 <rule id="100301" level="7">
  <if_sid>554</if_sid>
  <field name="file">/tmp/yara/malware/</field>
  <description>File added to /tmp/yara/malware/
directory.</description>
 </rule>
</group>

<group name="yara,">
 <rule id="108000" level="0">
  <decoded_as>yara_decoder</decoded_as>
  <description>Yara grouping rule</description>
 </rule>
 <rule id="108001" level="12">
  <if_sid>108000</if_sid>
  <match>wazuh-yara: INFO - Scan result: </match>
  <description>File "$(yara_scanned_file)" is a positive match. Yara rule:
$(yara_rule)</description>
 </rule>
</group>
```

2. Add the following decoders to the Wazuh server /var/ossec/etc/decoders/local_decoder.xml file.

This allows extracting the information from YARA scan results:

```
<decoder name="yara_decoder">
  <prematch>wazuh-yara:</prematch>
</decoder>

<decoder name="yara_decoder1">
  <parent>yara_decoder</parent>
  <regex>wazuh-yara: (\S+) - Scan result: (\S+)
(\S+)</regex>
  <order>log_type, yara_rule,
yara_scanned_file</order>
</decoder>
```

3. Add the following configuration to the Wazuh server /var/ossec/etc/ossec.conf configuration file. This configures the Active Response module to trigger after the rule 100300 and 100301 are fired:

```
<ossec_config>
  <command>
    <name>yara_linux</name>
    <executable>yara.sh</executable>
    <extra_args>-yara_path /usr/local/bin -yara_rules
/tmp/yara/rules/yara_rules.yar</extra_args>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <command>yara_linux</command>
    <location>local</location>
    <rules_id>100300,100301</rules_id>
  </active-response>
</ossec_config>
```

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

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

Attack emulation

1. Create the script /tmp/yara/malware/malware_downloader.sh on the monitored endpoint to download malware samples:

```
#!/bin/bash
# Wazuh - Malware Downloader for test purposes
# Copyright (C) 2015-2022, Wazuh Inc.
#
# This program is free software; you can redistribute it
# and/or modify it under the terms of the GNU General Public
# License (version 2) as published by the FSF - Free Software
# Foundation.

function fetch_sample(){

  curl -s -XGET "$1" -o "$2"

}

echo "WARNING: Downloading Malware samples, please use this script with  caution."
read -p "  Do you want to continue? (y/n)" -n 1 -r ANSWER
echo

if [[ $ANSWER =~ ^[Yy]$ ]]
then
   echo
   # Mirai
   echo "# Mirai: https://en.wikipedia.org/wiki/Mirai_(malware)"
   echo "Downloading malware sample..."
   fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/mirai" "/tmp/yara/malware/mirai"
&& echo "Done!" || echo "Error while downloading."
   echo

   # Xbash
   echo "# Xbash: https://unit42.paloaltonetworks.com/unit42-xbash-combines-botnet-ransomware-
coinmining-worm-targets-linux-windows/"
   echo "Downloading malware sample..."
   fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/xbash"
"/tmp/yara/malware/xbash" && echo "Done!" || echo "Error while downloading."
   echo

   # VPNFilter
   echo "# VPNFilter: https://news.sophos.com/en-us/2018/05/24/vpnfilter-botnet-a-sophoslabs-
analysis/"
   echo "Downloading malware sample..."
   fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/vpn_filter"
"/tmp/yara/malware/vpn_filter" && echo "Done!" || echo "Error while downloading."
   echo

   # Webshell
   echo "# WebShell: https://github.com/SecWiki/WebShell-
2/blob/master/Php/Worse%20Linux%20Shell.php"
   echo "Downloading malware sample..."
   fetch_sample "https://wazuh-demo.s3-us-west-1.amazonaws.com/webshell"
"/tmp/yara/malware/webshell" && echo "Done!" || echo "Error while downloading."
   echo
fi
```

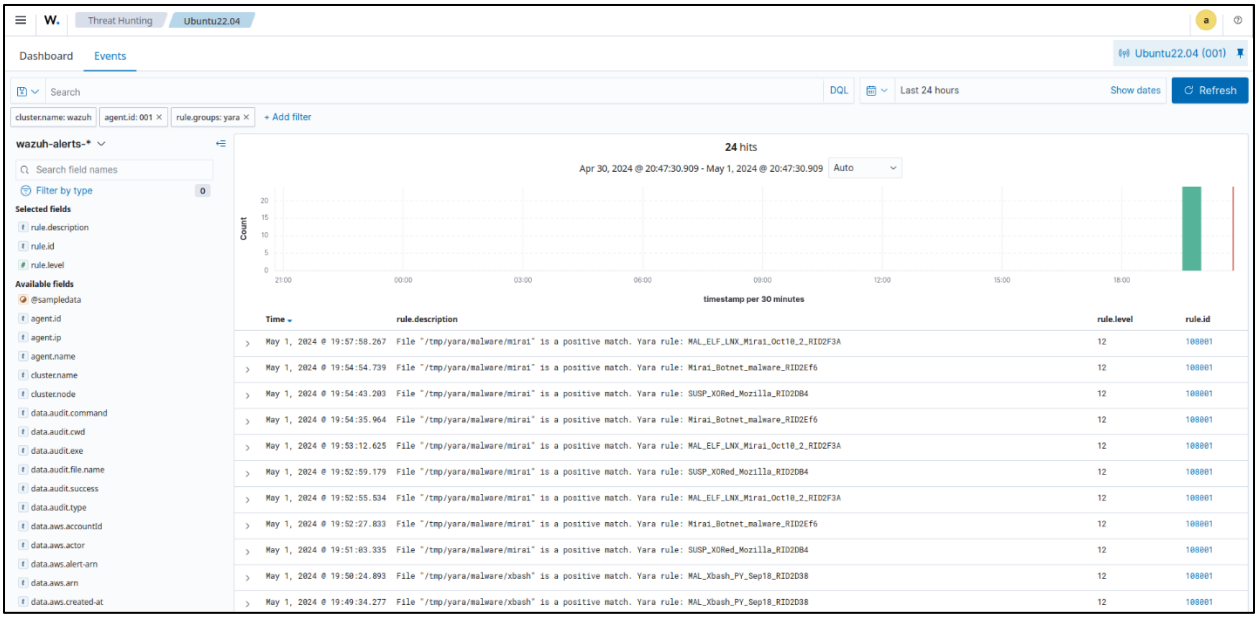
2. Run the malware_downloader.sh script to download malware samples to the /tmp/yara/malware directory:

```
sudo bash
/tmp/yara/malware/malware_downloader.sh
```

Visualize the alerts

You can visualize the alert data in the Wazuh dashboard. To do this, go to the **Threat Hunting** module and add the filters in the search bar to query the alerts.

rule.groups:yara

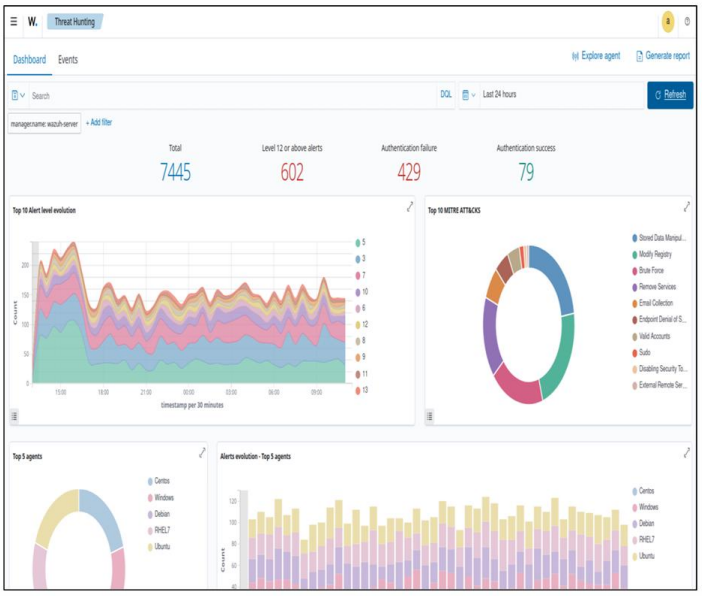


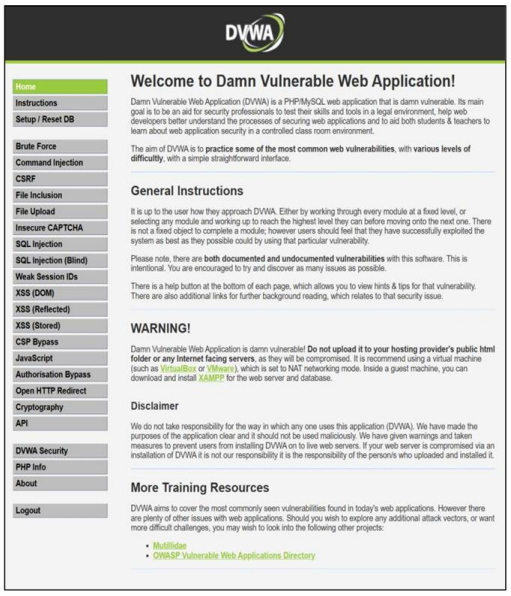
WAZUH TRAINING MODULE

Lab Training

Scenario 1: PHP Injection Attack

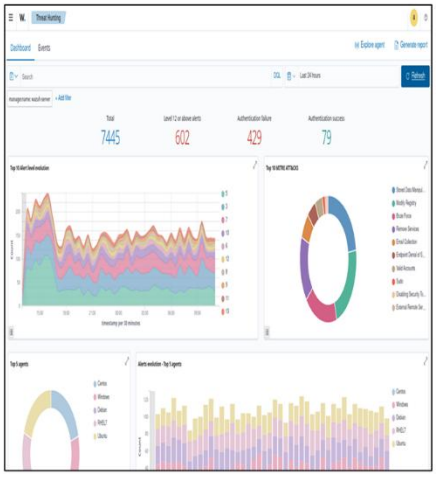
- Tools to be used Wazuh and DVWA as web server.

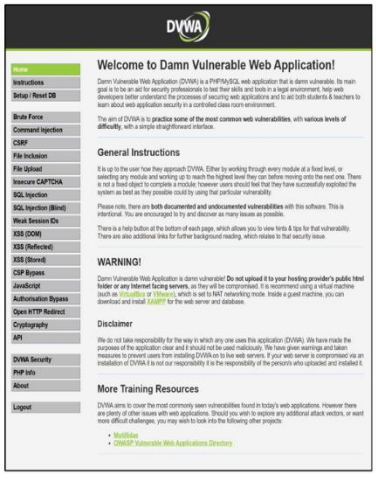


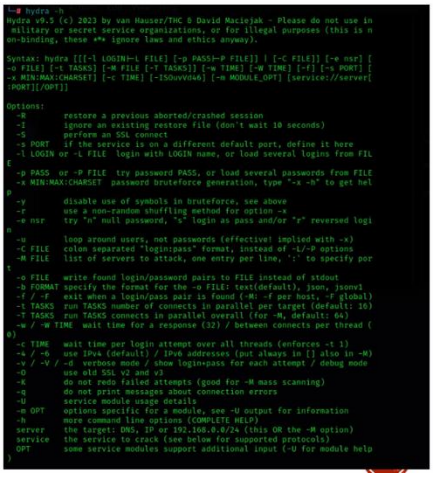


Scenario 2: Brute Force Attack

- Tools to be used is Wazuh, DVWA Web Server and Hydra as bruteforce tools in Kali Linux.







Scenario 3: Distributed Denial-of-Service (DDoS) attack

- Tools to be used is Wazuh, DVWA Web Server and MHDDOS as ddos tools in Kali Linux.