Wazuh Beginners to Advance

(For Ultimate Cloud Computing & Network Security)

By

Md. Jakaria

mjakaria.me

https://github.com/jakir-ruet

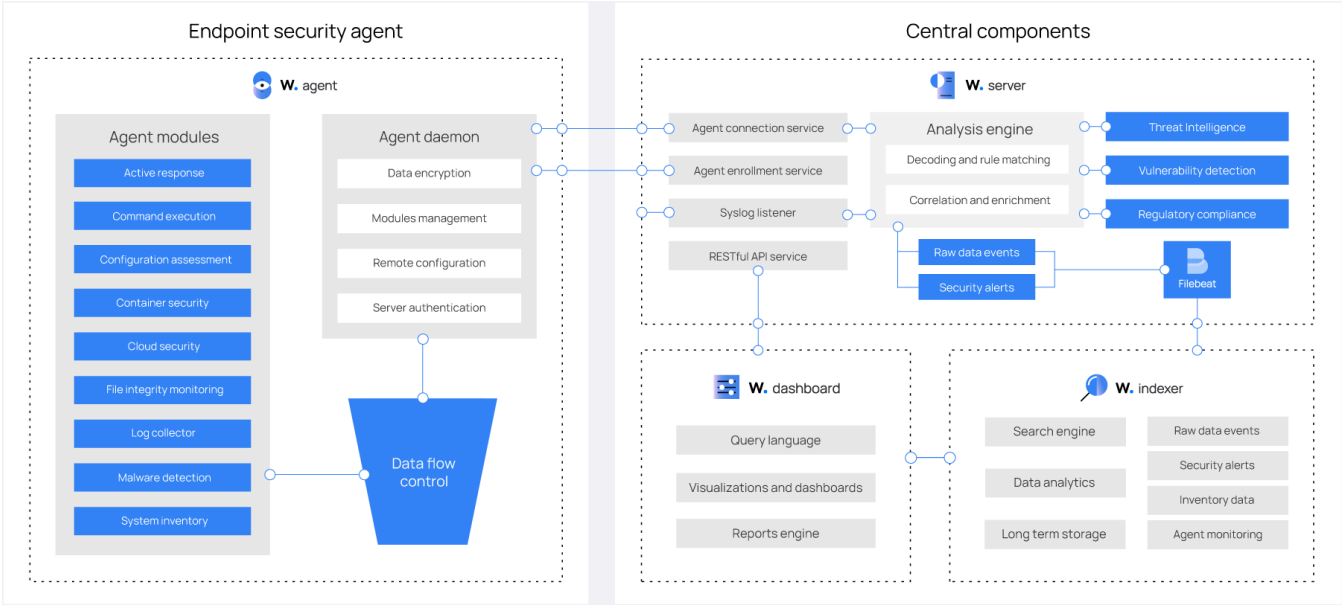
https://www.linkedin.com/in/jakir-ruet/

What is Wazuh?

Wazuh is an open-source security monitoring platform that provides log analysis, intrusion detection, vulnerability detection, and compliance monitoring. It can monitor your infrastructure for suspicious activity, configuration changes, and more.

Key Components of Wazuh

1. **Wazuh Manager:** The core component that performs analysis and decision-making.
2. **Wazuh Agents:** Installed on your systems (servers, endpoints) to collect and send data back to the Wazuh Manager.
3. **Elasticsearch:** Stores, indexes, and allows searching the log data (often integrated as part of the Elastic Stack).
4. **Kibana:** A visualization platform to display data and make reports from Elasticsearch.
5. **Filebeat:** Sometimes used to forward logs to Elasticsearch from agents.



Wazuh Indexer refers to the Elasticsearch component that is responsible for indexing and storing security data and alerts. Wazuh uses **Elasticsearch** as its backend data store to manage and search large volumes of security event data generated by the Wazuh Manager and its agents.

Key Roles of Wazuh Indexer (Elasticsearch):

1. **Data Indexing**:
   - Wazuh Indexer, built on **Elasticsearch**, indexes the incoming security event logs and alerts. This enables fast searches and quick retrieval of the event data, which is essential for real-time security monitoring and analysis.
2. **Storing Logs and Alerts**:
   - The Wazuh Indexer stores the security events and alerts generated by Wazuh agents and processed by the Wazuh Manager. These can include logs related to intrusion detection, system integrity, vulnerability detection, and more.
3. **Querying and Searching**:
   - The Wazuh Indexer provides powerful full-text search capabilities. This allows users to query large amounts of data for specific events, patterns, or anomalies. This is especially useful for incident response, compliance auditing, and threat detection.

4. **Data Retention**:
    - Elasticsearch manages the retention of indexed data over time. It allows Wazuh to manage how long data is stored before it is archived or deleted, which is important for maintaining storage efficiency and meeting compliance requirements.
5. **Scalability**:
    - Elasticsearch (and by extension, Wazuh Indexer) is highly scalable. As more data is collected from agents, the system can scale horizontally by adding more Elasticsearch nodes to handle increased loads.

Integration with Wazuh:

- **Wazuh** integrates with Elasticsearch to offer security monitoring capabilities, with **Wazuh Manager** sending data to the **Elasticsearch (Wazuh Indexer)** cluster.
- Alerts and event data are then stored in specific **Wazuh indices** in Elasticsearch (e.g., wazuh-alerts-*, wazuh-syscheck-*, etc.).
- The integration allows **Kibana**, another component of the Elastic Stack, to display security data from the **Wazuh Indexer** using interactive dashboards, making it easier for security teams to monitor and analyze system events.

Wazuh uses four different indices to store different event types:

Index	Description
wazuh-alerts	Stores alerts generated by the Wazuh server. These are created each time an event trips a rule with a high enough priority (this threshold is configurable).
wazuh-archives	Stores all events (archive data) received by the Wazuh server, whether or not they trip a rule.
wazuh-monitoring	Stores data related to the Wazuh agent status over time. It is used by the web interface to represent when individual agents are or have been Active, Disconnected, or Never connected.
wazuh-statistics	Stores data related to the Wazuh server performance. It is used by the web interface to represent the performance statistics.

Wazuh server

The Wazuh server component analyzes the data received from the agents, triggering alerts when threats or anomalies are detected. It is also used to manage the agents configuration remotely and monitor their status.

The Wazuh server uses threat intelligence sources to improve its detection capabilities. It also enriches alert data by using the MITRE ATT&CK framework and regulatory compliance requirements such as PCI DSS, GDPR, HIPAA, CIS, and NIST 800-53, providing helpful context for security analytics.

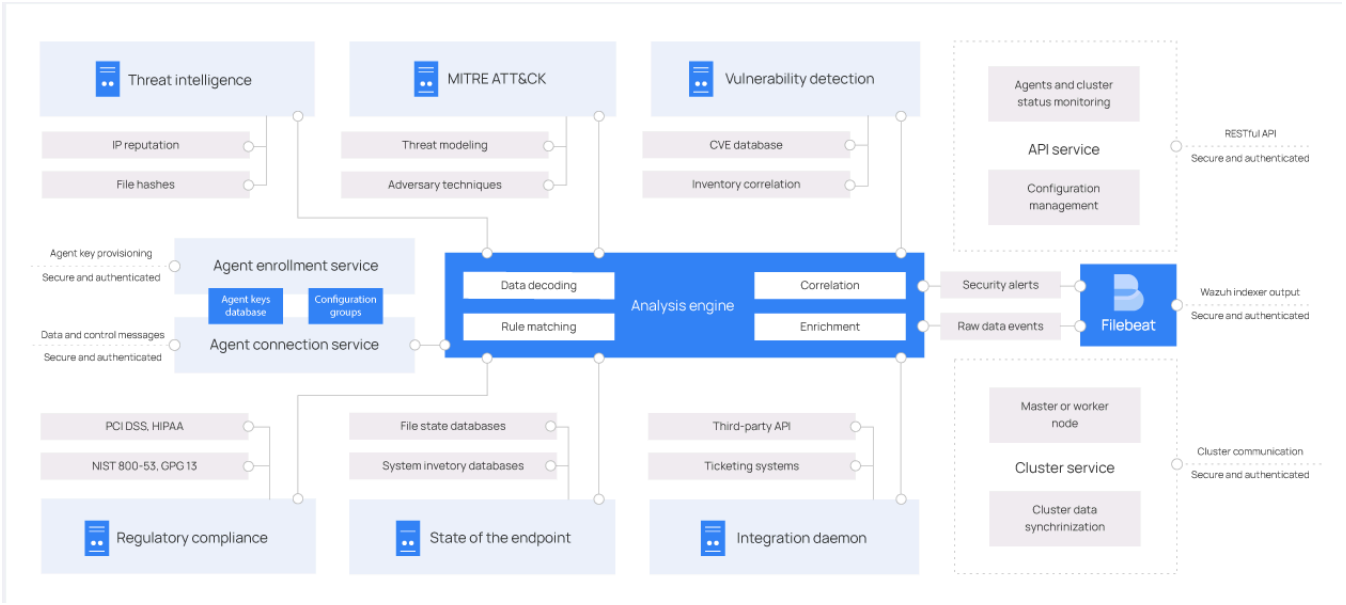
Additionally, the Wazuh server can be integrated with external software, including ticketing systems such as ServiceNow, Jira, and PagerDuty, as well as instant messaging platforms like Slack. These integrations are convenient for streamlining security operations.

Server architecture

The Wazuh server runs the analysis engine, the Wazuh RESTful API, the agent enrollment service, the agent

connection service, the Wazuh cluster daemon, and Filebeat. The server is installed on a Linux operating system and usually runs on a stand-alone physical machine, virtual machine, docker container, or cloud instance.

The diagram below represents the server architecture and components:



Server components

The Wazuh server comprises several components listed below that have different functions, such as enrolling new agents, validating each agent identity, and encrypting the communications between the Wazuh agent and the Wazuh server.

Agent enrollment service: It is used to enroll new agents. This service provides and distributes unique authentication keys to each agent. The process runs as a network service and supports authentication via TLS/SSL certificates or by providing a fixed password.

Agent connection service: This service receives data from the agents. It uses the keys shared by the enrollment service to validate each agent identity and encrypt the communications between the Wazuh agent and the Wazuh server. Additionally, this service provides centralized configuration management, enabling you to push new agent settings remotely.

Analysis engine: This is the server component that performs the data analysis. It uses decoders to identify the type of information being processed (Windows events, SSH logs, web server logs, and others). These decoders also extract relevant data elements from the log messages, such as source IP address, event ID, or username. Then, by using rules, the engine identifies specific patterns in the decoded events that could trigger alerts and possibly even call for automated countermeasures (e.g., banning an IP address, stopping a running process, or removing a malware artifact).

Wazuh RESTful API: This service provides an interface to interact with the Wazuh infrastructure. It is used to manage configuration settings of agents and servers, monitor the infrastructure status and overall health, manage and edit Wazuh decoders and rules, and query about the state of the monitored endpoints. The Wazuh

dashboard also uses it.

Wazuh cluster daemon: This service is used to scale Wazuh servers horizontally, deploying them as a cluster. This kind of configuration, combined with a network load balancer, provides high availability and load balancing. The Wazuh cluster daemon is what Wazuh servers use to communicate with each other and to keep synchronized.

Filebeat: It is used to send events and alerts to the Wazuh indexer. It reads the output of the Wazuh analysis engine and ships events in real time. It also provides load balancing when connected to a multi-node Wazuh indexer cluster.

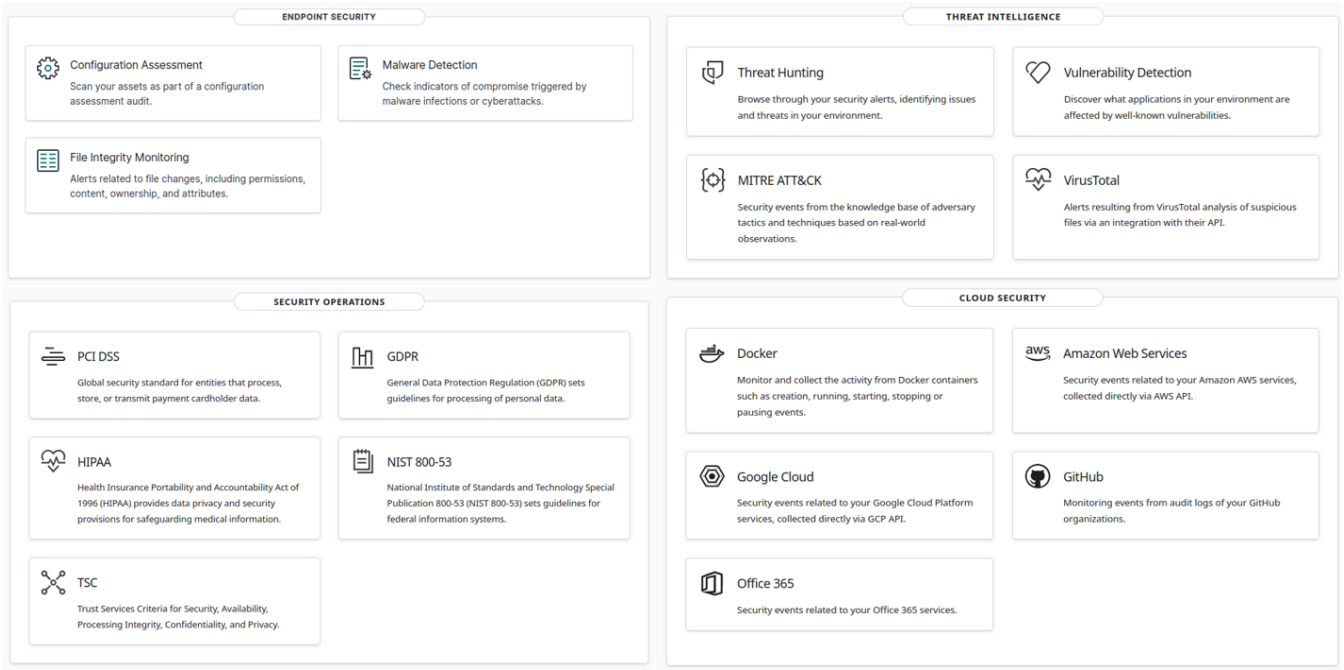
Wazuh dashboard

The Wazuh dashboard is a flexible and intuitive web user interface for mining, analyzing, and visualizing security events and alerts data. It is also used for the management and monitoring of the Wazuh platform. Additionally, it provides features for role-based access control (RBAC) and single sign-on (SSO).

Data visualization and analysis

The web interface helps users navigate through the different types of data collected by the Wazuh agent, as well as the security alerts generated by the Wazuh server. Users can also generate reports and create custom visualizations and dashboards.

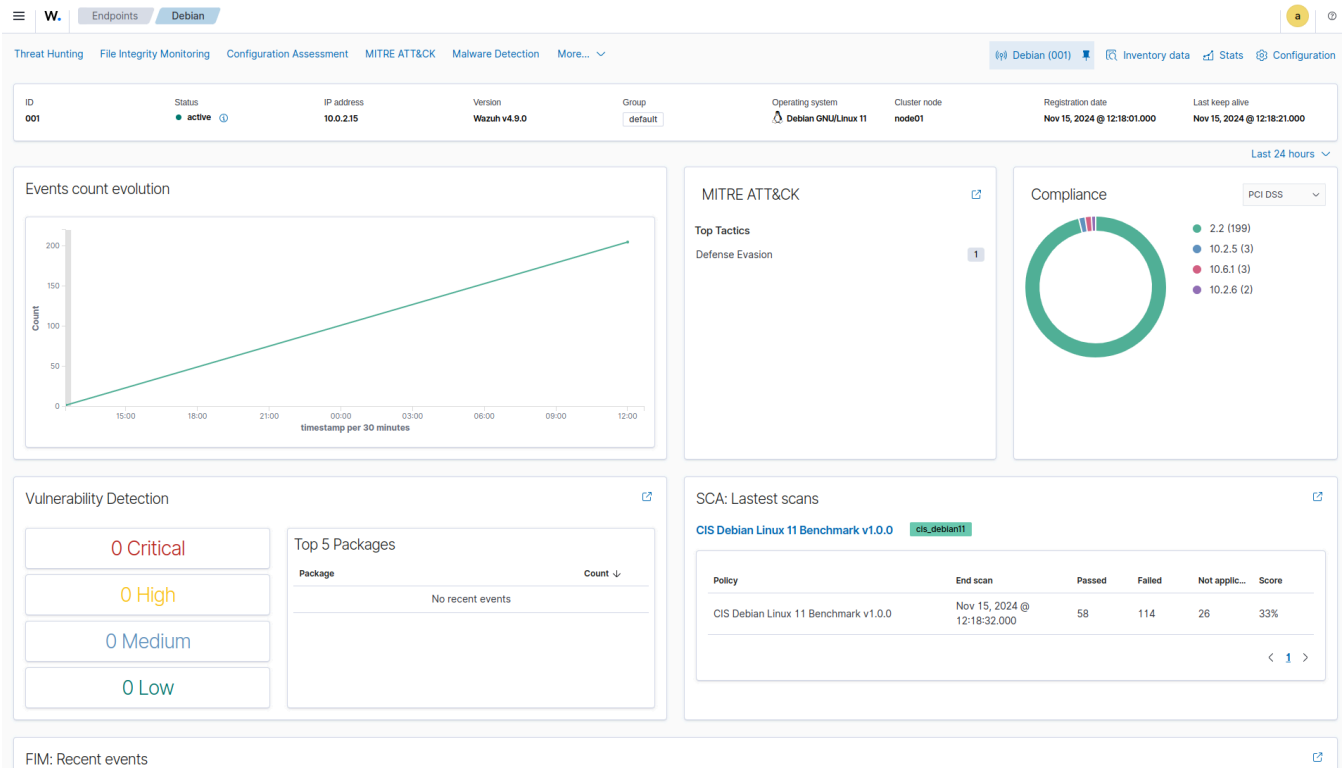
As an example, Wazuh provides out-of-the-box dashboards for regulatory compliance such as PCI DSS, GDPR, HIPAA, and NIST 800-53. It also provides an interface to navigate through the MITRE ATT&CK framework and related alerts.



Agents monitoring and configuration

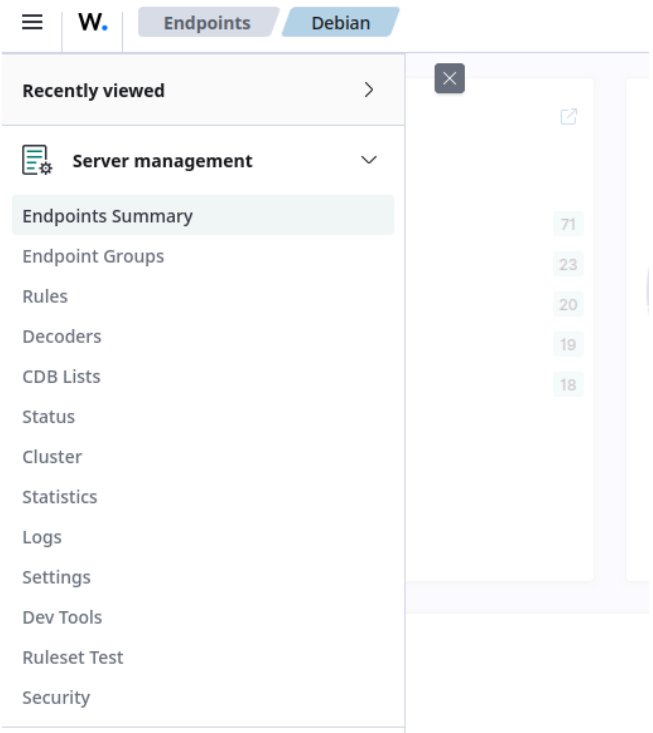
The Wazuh dashboard allows users to manage agents configuration and to monitor their status. As an example, for each monitored endpoint, users can define what agent modules will be enabled, what log files will be read,

what files will be monitored for integrity changes, or what configuration checks will be performed.



Platform management

The Wazuh dashboard provides a user interface dedicated to manage your Wazuh deployment. This includes monitoring the status, logs, and statistics of the different Wazuh components. It also includes configuring the Wazuh server, and creating custom rules and decoders for log analysis and threat detection.



Developer tools

The Wazuh dashboard includes a Ruleset Test tool that can process log messages to check how it is decoded and if it matches a threat detection rule or not. This feature is especially useful when custom decoders and rules have been created and the user wants to test them.

Ruleset test

The Wazuh dashboard also includes an API console for users to interact with the Wazuh API. This can be used to manage the Wazuh deployment (e.g., managing server or agent configurations, monitor status and log messages, adding or removing agents, etc.).

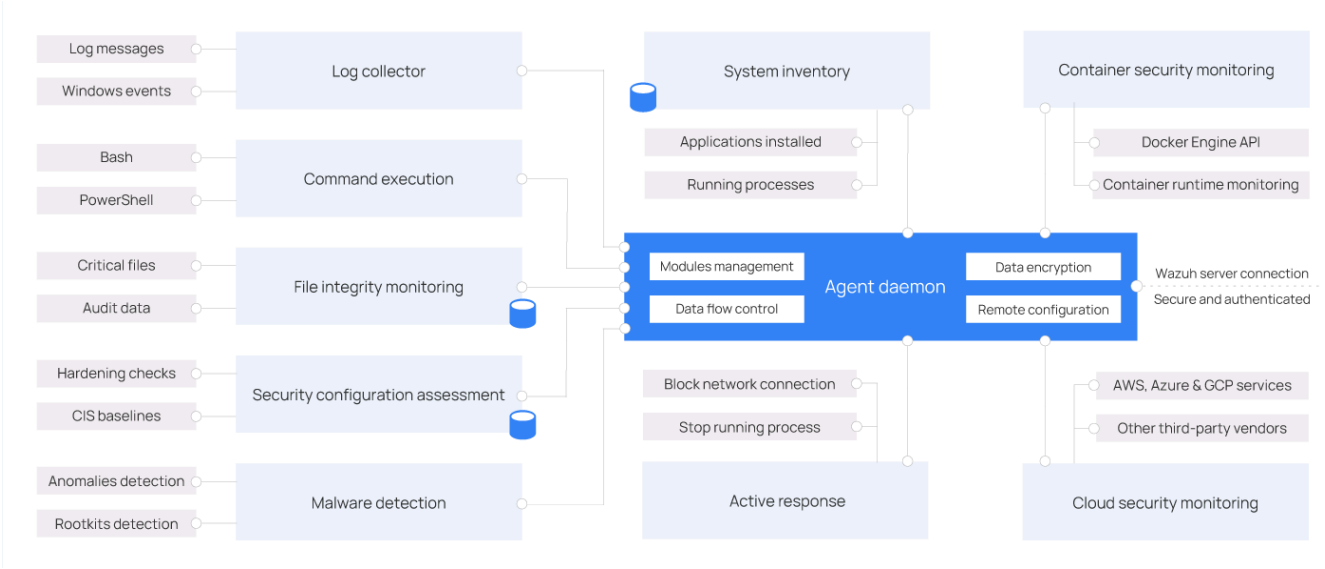
Wazuh agent

The Wazuh agent runs on Linux, Windows, macOS, Solaris, AIX, and other operating systems. It can be deployed to laptops, desktops, servers, cloud instances, containers, or virtual machines. The agent helps to protect your system by providing threat prevention, detection, and response capabilities. It is also used to collect different types of system and application data that it forwards to the Wazuh server through an encrypted and authenticated channel.

Agent architecture

The Wazuh agent has a modular architecture. Each component is in charge of its own tasks, including monitoring the file system, reading log messages, collecting inventory data, scanning the system configuration, and looking for malware. Users can manage agent modules via configuration settings, adapting the solution to their particular use cases.

The diagram below represents the agent architecture and components:



Agent modules

All agent modules are configurable and perform different security tasks. This modular architecture allows you to enable or disable each component according to your security needs. Below you can learn about the different purposes of all the agent modules.

Log collector: This agent component can read flat log files and Windows events, collecting operating system and application log messages. It supports XPath filters for Windows events and recognizes multi-line formats like Linux Audit logs. It can also enrich JSON events with additional metadata.

Command execution: Agents run authorized commands periodically, collecting their output and reporting it back to the Wazuh server for further analysis. You can use this module for different purposes, such as monitoring hard disk space left or getting a list of the last logged-in users.

File integrity monitoring (FIM): This module monitors the file system, reporting when files are created, deleted, or modified. It keeps track of changes in file attributes, permissions, ownership, and content. When an event occurs, it captures who, what, and when details in real time. Additionally, the FIM module builds and maintains a database with the state of the monitored files, allowing queries to be run remotely.

Security configuration assessment (SCA): This component provides continuous configuration assessment, utilizing out-of-the-box checks based on the Center of Internet Security (CIS) benchmarks. Users can also create their own SCA checks to monitor and enforce their security policies.

System inventory: This agent module periodically runs scans, collecting inventory data such as operating system version, network interfaces, running processes, installed applications, and a list of open ports. Scan results are stored in local SQLite databases that can be queried remotely.

Malware detection: Using a non-signature-based approach, this component is capable of detecting anomalies and the possible presence of rootkits. Also, it looks for hidden processes, hidden files, and hidden ports while monitoring system calls.

Active Response: This module runs automatic actions when threats are detected, triggering responses to block a network connection, stop a running process, or delete a malicious file. Users can also create custom responses when necessary and customize, for example, responses for running a binary in a sandbox, capturing network traffic, and scanning a file with an antivirus.

Container security monitoring: This agent module is integrated with the Docker Engine API to monitor changes in a containerized environment. For example, it detects changes to container images, network configuration, or data volumes. Besides, it alerts about containers running in privileged mode and about users executing commands in a running container.

Cloud security monitoring: This component monitors cloud providers such as Amazon Web Services, Microsoft Azure, or Google GCP. It natively communicates with their APIs. It is capable of detecting changes to the cloud infrastructure (e.g., a new user is created, a security group is modified, a cloud instance is stopped, etc.) and collecting cloud services log data (e.g., AWS Cloudtrail, AWS Macie, AWS GuardDuty, Azure Active Directory, etc.)

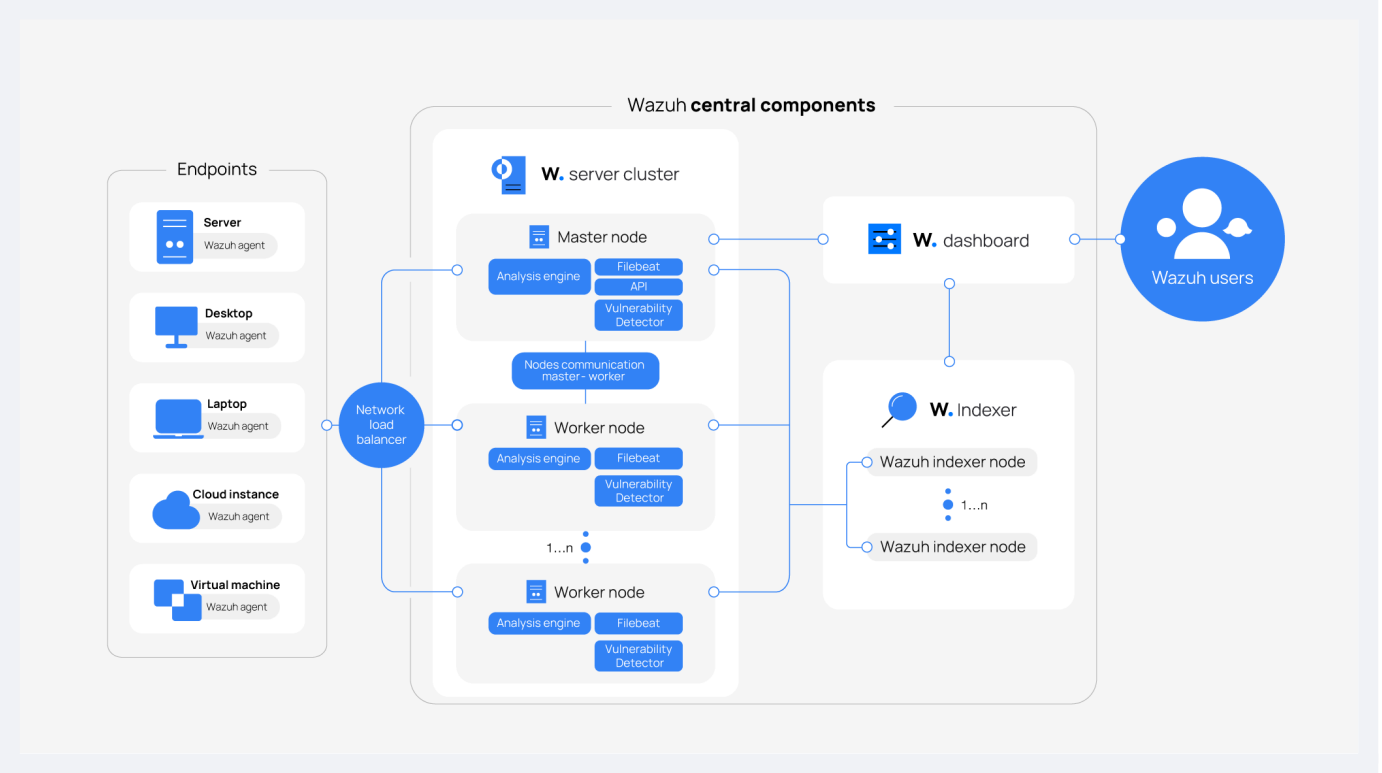
Communication with Wazuh server

The Wazuh agent communicates with the Wazuh server to ship collected data and security-related events. Besides, the agent sends operational data, reporting its configuration and status. Once connected, the agent can be upgraded, monitored, and configured remotely from the Wazuh server.

The communication of the agent with the server takes place through a secure channel (TCP or UDP), providing data encryption and compression in real time. Additionally, it includes flow control mechanisms to avoid flooding, queueing events when necessary, and protecting the network bandwidth.
You need to enroll the agent before connecting it to the server for the first time. This process provides the agent with a unique key used for authentication and data encryption.

Architecture

The Wazuh architecture is based on agents, running on the monitored endpoints, that forward security data to a central server. Agentless devices such as firewalls, switches, routers, and access points are supported and can actively submit log data via Syslog, SSH, or using their API. The central server decodes and analyzes the incoming information and passes the results along to the Wazuh indexer for indexing and storage.
The Wazuh indexer cluster is a collection of one or more nodes that communicate with each other to perform read and write operations on indices. Small Wazuh deployments, which do not require processing large amounts of data, can easily be handled by a single-node cluster. Multi-node clusters are recommended when there are many monitored endpoints, when a large volume of data is anticipated, or when high availability is required.
For production environments, it is recommended to deploy the Wazuh server and Wazuh indexer to different hosts. In this scenario, Filebeat is used to securely forward Wazuh alerts and archived events to the Wazuh indexer cluster (single-node or multi-node) using TLS encryption.
The diagram below represents a Wazuh deployment architecture. It shows the solution components and how the Wazuh server and the Wazuh indexer nodes can be configured as clusters, providing load balancing and high availability.



Wazuh agent - Wazuh server communication

The Wazuh agent continuously sends events to the Wazuh server for analysis and threat detection. To start shipping this data, the agent establishes a connection with the server service for agent connection, which listens on port 1514 by default (this is configurable). The Wazuh server then decodes and rule-checks the received events, utilizing the analysis engine. Events that trip a rule are augmented with alert data such as rule ID and rule name. Events can be spooled to one or both of the following files, depending on whether or not a rule is tripped:

- The file /var/ossec/logs/archives/archives.json contains all events whether they tripped a rule or not.
- The file /var/ossec/logs/alerts/alerts.json contains only events that tripped a rule with high enough priority (the threshold is configurable).

The Wazuh messages protocol uses AES encryption by default, with 128 bits per block and 256-bit keys. Blowfish encryption is optional.

Note Read the Benefits of using AES in the Wazuh communications document for more information.

Wazuh server - Wazuh indexer communication

The Wazuh server uses Filebeat to securely transmit alert and event data to the Wazuh indexer via TLS encryption. Filebeat monitors output data from the Wazuh server and forwards it to the Wazuh indexer, which listens on port 9200/TCP by default. Once indexed, you can analyze and visualize the data through the Wazuh dashboard.

The Vulnerability Detection module updates the vulnerability inventory. It also generates alerts, providing insights into system vulnerabilities.

The Wazuh dashboard queries the Wazuh RESTful API (by default listening on port 55000/TCP on the Wazuh server) to display configuration and status-related information of the Wazuh server and agents. It can also modify agents or server configuration settings through API calls. This communication is encrypted with TLS and authenticated with a username and password.

Required ports

Several services are used for the communication of Wazuh components. Below is the list of default ports used by these services. Users can modify these port numbers when necessary.

Component	Port	Protocol	Purpose
Wazuh server	1514	TCP (default)	Agent connection service
	1514	UDP (optional)	Agent connection service (disabled by default)
	1515	TCP	Agent enrollment service
	1516	TCP	Wazuh cluster daemon
	514	UDP (default)	Wazuh Syslog collector (disabled by default)
	514	TCP (optional)	Wazuh Syslog collector (disabled by default)
	55000	TCP	Wazuh server RESTful API
Wazuh indexer	9200	TCP	Wazuh indexer RESTful API
	9300-9400	TCP	Wazuh indexer cluster communication
Wazuh dashboard	443	TCP	Wazuh web user interface

Archival data storage

Both alerts and non-alert events are stored in files on the Wazuh server, in addition to being sent to the Wazuh indexer. These files can be written in JSON format (.json), or plain text format (.log). These files are daily compressed and signed using MD5, SHA1, and SHA256 checksums. The directory and filename structure is as follows:

root@wazuh-manager:/var/ossec/logs/archives/2022/Jan# ls -l

Output

```
total 176
-rw-r----- 1 wazuh wazuh 234350 Jan  2 00:00 ossec-archive-01.json.gz
-rw-r----- 1 wazuh wazuh    350 Jan  2 00:00 ossec-archive-01.json.sum
-rw-r----- 1 wazuh wazuh 176221 Jan  2 00:00 ossec-archive-01.log.gz
-rw-r----- 1 wazuh wazuh    346 Jan  2 00:00 ossec-archive-01.log.sum
-rw-r----- 1 wazuh wazuh 224320 Jan  2 00:00 ossec-archive-02.json.gz
-rw-r----- 1 wazuh wazuh    350 Jan  2 00:00 ossec-archive-02.json.sum
-rw-r----- 1 wazuh wazuh 151642 Jan  2 00:00 ossec-archive-02.log.gz
-rw-r----- 1 wazuh wazuh    346 Jan  2 00:00 ossec-archive-02.log.sum
-rw-r----- 1 wazuh wazuh 315251 Jan  2 00:00 ossec-archive-03.json.gz
-rw-r----- 1 wazuh wazuh    350 Jan  2 00:00 ossec-archive-03.json.sum
-rw-r----- 1 wazuh wazuh 156296 Jan  2 00:00 ossec-archive-03.log.gz
-rw-r----- 1 wazuh wazuh    346 Jan  2 00:00 ossec-archive-03.log.sum
```

Rotation and backups of archive files are recommended according to the storage capacity of the Wazuh server. By using cron jobs, you can easily manage to keep only a specific time window of archive files locally on the server, for example, last year or the last three months.

On the other hand, you may choose to dispense with storing archive files and simply rely on the Wazuh indexer for archive storage. This alternative might be preferred if you run periodic Wazuh indexer snapshot backups and/or have a multi-node Wazuh indexer cluster with shard replicas for high availability. You could even use a cron job to move snapshotted indices to a final data storage server and sign them using MD5, SHA1, and SHA256 hashing algorithms.

Use cases

The Wazuh platform helps organizations and individuals protect their data assets through threat prevention, detection, and response. Besides, Wazuh is also employed to meet regulatory compliance requirements, such as PCI DSS or HIPAA, and configuration standards like CIS hardening guides.

Moreover, Wazuh is also a solution for users of IaaS (Amazon AWS, Azure, or Google Cloud) to monitor virtual machines and cloud instances. This is done at a system level utilizing the Wazuh security agent and at an infrastructure level pulling data directly from the cloud provider API.

Additionally, Wazuh is employed to protect containerized environments by providing cloud-native runtime security. This feature is based on an integration with the Docker engine API and the Kubernetes API. The Wazuh security agent can run on the Docker host providing a complete set of threat detection and response capabilities.

Below you can find examples of some of the most common use cases of the Wazuh platform.

Endpoint security	Threat intelligence	Security operations	Cloud security
Configuration assessment	Threat hunting	Incident response	Container security
Malware detection	Log data analysis	Regulatory compliance	Posture management
File integrity monitoring	Vulnerability detection	IT hygiene	Workload protection

Setting Up Wazuh

- **Step 1: Install the Wazuh Manager:**
  Set up the central server that will receive and analyze data.
  Follow installation guides for different operating systems like Linux, Windows, or Docker.
- **Step 2: Install Wazuh Agent:**
  Set up the agent on the systems you want to monitor (e.g., your servers or workstations).
- **Step 3: Configure Communication:**
  Make sure that the Wazuh Manager and agents can communicate securely.
- **Step 4: Connect to Elasticsearch and Kibana:**
  To visualize your log data in Kibana, ensure that the Wazuh manager integrates properly with the Elastic Stack.

Basic Wazuh Features

- **Log Collection:**
  Wazuh agents collect and forward logs from various sources like system logs, application logs, firewalls, and more.
- **Intrusion Detection:**
  Using its rules engine, Wazuh identifies potential threats such as brute force attacks, malware, and

unauthorized changes.

- **File Integrity Monitoring:**
  Monitor changes to critical system files, directories, and configurations.
- **Vulnerability Detection:**
  Identify known vulnerabilities based on information from software and configuration databases.
- **Security Incident Response:**
  Automatically trigger responses to certain events (e.g., blocking IPs, sending alerts).

Wazuh Rules and Decoders

- **Decoders:** Convert raw log data into structured information.
- **Rules:** A set of pre-configured conditions that check if logs contain signs of threats or policy violations. Wazuh includes predefined rules for a wide range of events such as failed login attempts, buffer overflows, and more.

Monitoring and Reporting with Kibana

- **Kibana Dashboards:**
  After setting up Elasticsearch and Kibana, Wazuh provides a set of pre-configured dashboards to visualize security-related data like intrusion attempts, authentication failures, and vulnerabilities.
- **Alerts:**
  When Wazuh detects suspicious activity, it generates alerts. These alerts can be reviewed in Kibana, and you can take actions accordingly.

Common Use Cases for Wazuh

- **Compliance:** Wazuh helps in meeting compliance standards like HIDS, PCI DSS, HIPAA, GDPR, and more by monitoring systems and generating audit reports.
- **Incident Detection and Response:** Automate responses and manage incidents related to security breaches.
- **Vulnerability Management:** Keep track of vulnerabilities in your systems and take necessary action based on their severity.

Brief in Compliance Monitoring:

1. HIDS (Host Intrusion Detection System)

- A Host Intrusion Detection System (HIDS) is a type of security software that monitors and analyzes the internal activities of a computer or network device to detect suspicious behavior or potential security threats. Unlike Network Intrusion Detection Systems (NIDS), which monitor network traffic, HIDS operates on individual host machines (such as servers or workstations) to watch for signs of intrusion, malware, or other abnormal activities.
- **Key Functions**:
  - Monitors system logs, file integrity, and configuration changes.
  - Detects unauthorized access or activities (e.g., file modifications, system calls).
  - Provides alerts or automated responses when an anomaly is detected.

2. GDPR (General Data Protection Regulation)

- The General Data Protection Regulation (GDPR) is a regulation enacted by the European Union (EU) in 2018 to protect the personal data and privacy of EU citizens. It sets strict rules for how organizations

collect, process, store, and share personal data, with a strong emphasis on transparency, accountability, and individuals' control over their own data.

- **Key Principles**:
    - **Data Minimization**: Only collect the minimum amount of personal data necessary for the purpose.
    - **Consent**: Organizations must obtain explicit consent from individuals to process their personal data.
    - **Right to Access & Deletion**: Individuals have the right to access their personal data and request its deletion.
    - **Data Protection by Design**: Organizations must implement data protection measures in their processes and systems from the outset.
    - **Breach Notification**: Organizations must report data breaches to the relevant authorities within 72 hours.

3. HIPAA (Health Insurance Portability and Accountability Act)

- HIPAA is a U.S. law designed to protect the privacy and security of individuals' health information. It sets standards for the handling of Protected Health Information (PHI) by healthcare organizations, insurers, and their business associates.
- **Key Components**:
    - **Privacy Rule**: Sets national standards for the protection of health information.
    - **Security Rule**: Establishes requirements for safeguarding electronic health information (ePHI).
    - **Breach Notification Rule**: Requires healthcare providers and entities to notify individuals if their PHI is breached.
    - **Penalties**: Violations of HIPAA can lead to hefty fines, depending on the severity of the breach.

4. PCI-DSS (Payment Card Industry Data Security Standard)

- PCI-DSS is a set of security standards designed to protect cardholder data and secure payment card transactions. It applies to all organizations that process, store, or transmit payment card information (credit, debit, etc.). Compliance with PCI-DSS is required to ensure that sensitive payment data is protected from breaches or theft.
- **Key Requirements**:
    - **Encryption**: Payment card information must be encrypted when stored or transmitted.
    - **Access Control**: Limiting access to cardholder data to only those who need it.
    - **Network Security**: Implementing firewalls, intrusion detection systems, and secure configurations to protect cardholder data.
    - **Regular Monitoring**: Monitoring systems and processes for vulnerabilities and breaches.
    - **Security Testing**: Regular vulnerability assessments, penetration testing, and system audits to identify weaknesses.

Wazuh Documentation and Learning Resources

- **Official Documentation:** The Wazuh Documentation is the best place to start. It covers everything from installation to advanced configuration.
- **Wazuh Community:** Join the Wazuh Community for discussions, forums, and open-source collaboration.
- **YouTube Tutorials:** There are various video tutorials available on YouTube that walk through different aspects of Wazuh, from installation to configuration and rule creation.

Advanced Topics

- **Custom Rules:** Create your own rules to detect specific behaviors or events tailored to your environment.
- **Integrations:** Wazuh can integrate with third-party systems, such as SIEMs or security tools, for enhanced functionality.
- **Scalability:** Learn how to scale Wazuh for larger environments with multiple managers, distributed agents, and centralized storage.

Open Source Security (OSSEC):

OSSEC (Open Source Security) is a free, open-source Host Intrusion Detection System (HIDS) that is widely used for monitoring and securing systems and networks. It is designed to detect potential security threats, such as intrusions, malware, configuration changes, and other suspicious activity, by analyzing log files and system behavior.

Key Features of OSSEC:

1. **Log Analysis**:
    - OSSEC analyzes log data from various sources (e.g., system logs, application logs, and firewall logs) to detect abnormal activities that could indicate a security threat.
    - It uses a set of predefined rules and custom rules to correlate and identify potential threats.
2. **File Integrity Monitoring**:
    - OSSEC monitors critical files and directories on your systems and checks for any unauthorized changes.
    - It can alert administrators if there are any unexpected modifications to files, which is often a sign of malware or a security breach.
3. **Rootkit Detection**:
    - OSSEC includes features for detecting rootkits, which are tools or software that attackers use to gain unauthorized access to a system while concealing their presence.
    - It can detect hidden processes, files, and kernel-level changes that indicate the presence of rootkits.
4. **Active Response**:
    - OSSEC can be configured to automatically respond to certain security events by taking actions like blocking IP addresses, shutting down compromised services, or running predefined scripts.
    - This is useful for mitigating threats without waiting for manual intervention.
5. **Real-Time Alerts**:
    - OSSEC generates real-time alerts when suspicious behavior or security breaches are detected.
    - Alerts can be sent via email, syslog, or integrated with other systems for further analysis and response.
6. **Intrusion Detection**:
    - OSSEC is capable of detecting various types of intrusions, including unauthorized access attempts, brute-force attacks, and unauthorized privilege escalation.
    - It can monitor both Windows and Linux/Unix systems.
7. **Scalability**:
    - OSSEC is designed to scale from small systems to large enterprise networks.
    - It has a central management server, making it easier to monitor multiple hosts and devices across an organization.

8. **Configuration Compliance**:
    - OSSEC can also help organizations ensure compliance with security standards (e.g., PCI-DSS, HIPAA, and GDPR) by auditing and reporting on system configurations and security settings.
    - It can generate reports on configuration compliance, identifying areas that might need improvement.

OSSEC vs. Wazuh:

Feature	OSSEC	Wazuh
Overview	Open-source Host-based Intrusion Detection System (HIDS).	A fork of OSSEC with additional features, including better scalability, reporting, and integration with the Elastic Stack.
Main Focus	Host Intrusion Detection, Log analysis, File integrity monitoring.	Host Intrusion Detection, Log analysis, File integrity monitoring, and extended features like advanced security monitoring and SIEM integration.
Fork of OSSEC	Yes, OSSEC is the predecessor.	Yes, Wazuh is a fork of OSSEC but with enhancements.
Real-time Alerts	Yes, alerts on suspicious activity or security events.	Yes, enhanced real-time alerts with customizable notifications.
Log Management	Analyzes and correlates logs to detect anomalies.	More advanced log management, integrates with Elasticsearch, Logstash, and Kibana (ELK Stack) for enhanced visualization and analytics.
File Integrity Monitoring (FIM)	Yes, detects changes in important system files.	Yes, with enhanced capabilities and better reporting features.
Rootkit Detection	Yes, detects and alerts on potential rootkit activity.	Yes, with improved detection and integration into centralized security operations.
Scalability	Suitable for small to medium-sized environments.	Designed for larger environments and is more scalable, especially when integrated with the ELK stack.
Compliance Reporting	Can help with compliance, but reporting features are basic.	Advanced compliance reporting with features for standards like PCI-DSS, HIPAA, GDPR, and others.
Active Response	Yes, can trigger automatic responses such as blocking IPs or running commands.	Yes, enhanced active response capabilities, can also interact with external systems for automatic remediation.
Web Interface	No built-in web interface; uses command-line and log files.	Yes, includes a built-in web interface for easier management and visualization.
Integration with Other Tools	Limited to basic integrations.	Strong integration with SIEM tools, including the Elastic Stack (Elasticsearch, Logstash, Kibana), making it ideal for centralizing security data.
Ease of Use	Requires more manual setup and configuration.	More user-friendly with improved configuration and management interfaces.
Community and	Strong community but lacks	Larger community with active development and

Support	extensive enterprise-level support.	enterprise-level support options available.
Updates and Features	Updates are less frequent.	Actively maintained with frequent updates and new features.

Key Takeaways:

- **OSSEC** is a lighter, simpler tool suitable for smaller environments or those with a focus on traditional host-based intrusion detection and log analysis.
- **Wazuh** is a more feature-rich and scalable solution built on OSSEC's foundation. It adds advanced features like SIEM integration (with Elastic Stack), better compliance reporting, and a more user-friendly web interface, making it more suitable for larger environments and modern security operations.

Hands-On Practice

- Set up a small test environment with Wazuh Manager and a few agents.
- Simulate some common attacks (e.g., brute force, malware activity) to see how Wazuh detects and alerts you.
- Explore Kibana dashboards to get a feel for the data presentation.