wazuh.

Advanced Training for Security Engineers

› 4-Day Course ‹

Wazuh Advanced Training for Security Engineers **› 4-Day Course ‹**

Overview

Version 1.0 — June 2026

This four-day advanced training course is designed for security engineers and SOC professionals with prior hands-on experience with Wazuh. Students will configure advanced authentication, deploy and manage Wazuh platform modules, integrate external threat intelligence sources, administer the platform via API, and implement AI-assisted security operations workflows.

This course consists of lectures and hands-on exercises performed in a virtual lab environment provided by our team. The exercises guide students through advanced configuration, API-driven operations, threat intelligence integration, and AI-assisted security workflows. Students are expected to have prior Wazuh experience and comfort with Linux administration. Comprehensive course materials will be provided, with lab access remaining available for 24 hours after the course ends.

Course Materials Notice: This syllabus reflects the course structure and topic coverage as of the publication date. To ensure content relevance and quality, individual lab exercises may be updated or substituted prior to or during course delivery. Core topics and learning objectives remain consistent across all sessions.

4 days	Wazuh 4.14.x
DURATION	SOFTWARE VERSIONS

Intended audience

- Security engineers with prior Wazuh experience
- SOC analysts and Wazuh administrators
- IT professionals responsible for advanced Wazuh configuration and operations

Prerequisites:

Completion of Wazuh Essential Training or equivalent hands-on Wazuh experience.

Familiarity with core Wazuh components: manager, agent, dashboard, and indexer.

*Networking should allow outgoing traffic for port 22 (SSH), 3389 (RDP), and 443 (UI) to *.lab.wazuh.info*

Course Objectives

This course is designed to develop the following capabilities in participants:

01 Configure advanced authentication, including LDAP, SAML SSO, and MFA.

02 Manage RBAC roles, users, and permissions across multi-tenant SOC environments.

03 Deploy and manage Wazuh wodles for system inventory, cloud log ingestion, and container monitoring.

04 Build and customize Wazuh integrations with third-party platforms and threat intelligence feeds.

05 Implement IoC-based detection using CDB lists, MISP, VirusTotal, and YARA.

06 Administer Wazuh via the Manager API and Wazuh-indexer API for operational automation.

07 Design and manage index lifecycle policies, ingest pipelines, and reindex workflows.

08 Configure advanced FIM with real-time monitoring, baselining, and Active Response.

Table of Contents

Session 1

Platform Security and Authentication Management

This module covers the Wazuh platform's authentication and access control architecture. Students will configure role-based access control, connect external identity providers, and implement advanced credential management strategies.

This module is designed to cover the following objectives:

- Understand how authentication and authorization are delegated to the Wazuh-indexer security plugin.
- Configure Dashboard access with TLS enforcement and protected endpoints.
- Create and manage internal users and roles using the Security plugin.
- Implement RBAC with least-privilege role mappings for index-level access control.
- Understand LDAP and Active Directory as external authentication domains.
- Understand SAML Single Sign-On with external Identity Providers.
- Implement advanced password management strategies for Wazuh system accounts.
- Understand multi-factor authentication (MFA) for Wazuh Dashboard access.

Representative lab exercises for this module include:

- Lab 1A: Configure TLS enforcement, create a read-only SOC analyst role, and validate index-level access restrictions.
- Lab 1B: Create and map roles for two distinct teams and validate per-user data scope restrictions.

Wazuh Modules and Platform Capabilities

This module covers Wazuh's modular data collection components (wodles) and their configuration for system inventory, cloud log ingestion, container monitoring, and Kubernetes audit visibility.

This module is designed to cover the following objectives:

- Understand the Wodle architecture and its role in extending Wazuh agent capabilities.
- Configure the syscollector wodle for hardware, OS, and package inventory collection.
- Deploy the aws-s3 wodle to ingest CloudTrail and other AWS logs into Wazuh.
- Understand the azure-logs wodle for Azure Log Analytics and Active Directory audit ingestion.
- Build and deploy custom wodles using the wodle-command module.
- Enable and validate the Docker-listener module for container event monitoring.
- Understand Kubernetes API audit log collection through a webhook listener.

Representative lab exercises for this module include:

- Lab 2A: Configure and validate the syscollector wodle for system inventory collection.
- Lab 2B: Configure the aws-s3 wodle to ingest AWS CloudTrail logs into Wazuh.
- Lab 2C: Build and deploy a custom command wodle for scheduled data collection.
- Lab 2D: Enable the docker-listener wodle and validate container lifecycle event alerts.
- Lab 2E: Deploy a Kubernetes API audit webhook listener and create detections for key API verbs.

Session 2

Integrations and Log Ingestion Troubleshooting

This module covers the Wazuh Integrator feature and its use for forwarding selected alerts to external platforms. Students will configure built-in and custom integrations, design log ingestion workflows, and troubleshoot common ingestion and capacity issues.

This module is designed to cover the following objectives:

- Understand how the Wazuh Integrator operates as a post-detection alert forwarding module.
- Configure built-in integrations with Slack, Maltiverse, and AbuseIPDB.
- Build custom integration scripts for third-party platforms and threat intelligence APIs.
- Understand log path prefixes and ingestion workflow design.
- Understand remote syslog ingestion and log forwarding into Wazuh.
- Troubleshoot dropped events, EPS limits, and agent queue behavior.
- Understand and design Wazuh-indexer architectures to prevent bottlenecks and shard allocation issues.

Representative lab exercises for this module include:

- Lab 3A: Configure a production-ready Slack integration with level-based alert filtering.
- Lab 3B: Configure the Maltiverse integration and validate multi-type IoC enrichment.
- Lab 3C: Build and deploy a custom AbuseIPDB integration script for SSH alert enrichment.

Threat Intelligence and IoC-Driven Detection

This module covers how Wazuh integrates with external threat intelligence sources to enable deterministic IoC-based detection. Students will configure MISP feed ingestion, VirusTotal enrichment, and YARA-based malware scanning within the Wazuh detection pipeline.

This module is designed to cover the following objectives:

- Understand the distinction between IoC-based detection, threat hunting, and malware detection in Wazuh.
- Use MITRE ATT&CK context to prioritize, investigate, and communicate detections.
- Build IoC detection rules using CDB lists and custom decoder and rule pairs.
- Configure MISP feed ingestion and automate IoC list refresh via the Manager API.
- Integrate VirusTotal for hash-based file reputation enrichment.
- Deploy YARA rules for malware artifact classification on monitored endpoints.
- Implement an IoC lifecycle management and detection strategy.

This module covers the following topics:

- Scope and terminology: IoC detection, threat hunting, malware detection, and MITRE ATT&CK mapping.
- CDB list lookups for deterministic indicator matching in rules.
- MISP integration: feed ingestion, list automation, and supported IoC types.
- VirusTotal integration: file hash submission and reputation enrichment.
- YARA integration: rule deployment and malware classification workflows.
- IoC lifecycle management strategy including feed refresh automation and indicator expiration.

Representative lab exercises for this module include:

- Lab 4A: Create a CDB list of known malicious IPs, write a detection rule, and validate end-to-end alert generation.
- Lab 4B: Automate MISP feed ingestion, refresh IoC lists via the Manager API, and validate deterministic IoC matches.

Session 3

Wazuh Cloud Administration and API Usage

This module covers programmatic interaction with the Wazuh platform using the Manager API and Wazuh-indexer API. Students will perform operational validation, design data extraction workflows, and manage index lifecycle policies, templates, and ingest pipelines.

This module is designed to cover the following objectives:

- Authenticate to the Wazuh Manager API using JWT and perform operational validations.
- Validate Wazuh Cloud control-plane availability using the Cloud API.
- Generate temporary cold storage credentials for controlled archive retrieval.
- Retrieve agent connectivity metrics and manager runtime status via API.
- Implement validation gates and secure secret handling for API automation workflows.
- Create and apply ISM policies for automated index lifecycle management.
- Build ingest pipelines for field normalization and conditional data transformation.
- Perform reindex operations for schema migrations and data consolidation.
- Build a reusable API health reporting script for SOC operational handoffs.

Representative lab exercises for this module include:

- Lab 5A: Validate Wazuh Cloud control-plane availability and extract deployment metadata via the Cloud API.
- Lab 5B: Generate and validate temporary cold storage access credentials.
- Lab 5C: Authenticate to the Manager API, validate runtime services, and retrieve agent connectivity metrics.
- Lab 5D: Implement API validation gates, troubleshoot common error codes, and apply secure secret handling patterns.
- Lab 5E: Create an index template and ISM policy and validate automated index lifecycle state transitions.
- Lab 5F: Build an OpenSearch ingest pipeline with field normalization and validate pipeline behavior at ingestion time.

- Lab 5G: Reindex historical alert data into a destination index and validate document count and field consistency.
- Lab 5H: Build a reusable API health reporting script that aggregates manager status, agent state, and top rules.

File Integrity Monitoring and Active Response

This module covers the internal architecture of Wazuh's File Integrity Monitoring (FIM) component and its integration with Active Response. Students will configure advanced FIM tuning, cross-platform monitoring, and automated remediation workflows triggered by file change detection.

This module is designed to cover the following objectives:

- Understand the internal FIM architecture including the Syscheck module, baseline management, and real-time monitoring.
- Differentiate FIM behavior across Linux (inotify), Windows (ReadDirectoryChangesW), and macOS (FSEvents).
- Understand inode changes and sentinel values in FIM alerts for accurate change attribution.
- Balance monitoring coverage with system performance through strategic path selection.
- Design FIM correlation scenarios for multi-stage threat detection.
- Configure Active Response to trigger automated remediation based on FIM alert conditions.
- Integrate YARA with FIM to scan newly written or modified files for malware artifacts.
- Integrate VirusTotal with FIM for hash-based reputation lookup and automated file removal.
- Build and validate explicit FIM baselines for critical directories.

Representative lab exercises for this module include:

- Lab 6A: Configure YARA rule scanning triggered by FIM events and validate Active Response execution on detected malware.
- Lab 6B: Configure VirusTotal integration with FIM to submit file hashes and trigger automated removal of confirmed malicious files.
- Lab 6C: Build an explicit FIM baseline for a critical directory, identify unauthorized changes, and trigger Active Response.

Session 4

Dashboards and Reporting for SOC Operations

This module covers the design, deployment, and operational use of Wazuh dashboards for multi-tenant SOC workflows. Students will build custom visualizations, generate PDF reports, and configure dashboard branding for enterprise and MSP deployments.

This module is designed to cover the following objectives:

- Design reusable dashboards that support multi-tenant SOC triage and investigation workflows.
- Understand RBAC and index-level permissions to enforce data isolation across tenants.
- Build custom visualizations, including metrics widgets, donut charts, and interactive data tables.
- Structure dashboards in investigation layers from alert summary to event-level drill-down.
- Generate and deliver PDF reports directly from the Wazuh Dashboard.
- Implement dashboard branding customization for organizational or customer-facing deployments.

Representative lab exercises for this module include:

- Lab 7A: Build a Vulnerabilities Report Dashboard from scratch with seven visualization panels.
- Lab 7B: Generate a filtered PDF report of critical alerts from the past 24 hours and deliver to stakeholders.
- Lab 7C: Apply custom branding to the Wazuh Dashboard including logo and color customization.

AI Assistance in Security Operations

This module covers the integration of AI-assisted capabilities into Wazuh security workflows. Students will configure alert enrichment via external AI APIs, implement alert summarization pipelines, and design human-gated Active Response workflows.

This module is designed to cover the following objectives:

- Understand the role of AI in augmenting SOC analyst workflows without replacing human oversight.
- Configure a custom Wazuh integration script that forwards critical alerts to an external AI API.
- Deliver AI-generated alert summaries to a SOC communication channel via Slack.
- Implement output validation for AI integration scripts to prevent pipeline failures.
- Apply AI-generated enrichment to classify alert risk in near real time.
- Design governance boundaries for AI-assisted workflows including human approval gates.
- Understand operational risks of AI-assisted automated remediation.

Representative lab exercises for this module include:

- Lab 8A: Build a custom integration that forwards critical alerts to ChatGPT for plain-language summarization and delivers results to a SOC Slack channel.
- Lab 8B: Extend the AI integration to classify risk and implement a human approval gate before triggering Active Response.