History of HIDS, OSSEC, and Wazuh

1. What is HIDS?

HIDS = Host-based Intrusion Detection System

Monitors individual systems (hosts) for suspicious activity like unauthorized access, file changes, malware, etc.

🔍 **Key Features of HIDS:**

- Log analysis
- File integrity monitoring (FIM)
- Rootkit detection
- Real-time alerting

🕰️ **HIDS Origins:**

Era	Milestone/Description
1980s–90s	Initial IDS research in academic/security labs
Mid-1990s	Early HIDS tools emerged (e.g., Tripwire, Dragon Squire)
2000s	Open-source & commercial HIDS tools gained popularity

2. OSSEC – Open Source Security Event Correlator

📌 **Introduction:**

- **Released**: 2004 by **Daniel B. Cid**
- **Language**: C
- **License**: GPL
- One of the first **open-source HIDS** tools with real-time detection & centralized management.

🔧 **Core Capabilities:**

- Multi-platform (Linux, Windows, macOS, BSD)
- Log analysis
- File integrity monitoring (FIM)
- Rootkit detection
- Active response (firewall rules, etc.)
- Agent-based architecture with central manager

Timeline:

Year	Event
2004	OSSEC project launched by Daniel B. Cid
2008	Acquired by **Trend Micro**
2010s	Maintained slowly; fewer updates and enhancements

3. Wazuh – Fork of OSSEC (Modern HIDS/EDR/XDR)

📌 **Introduction:**

- **Founded**: 2015 as a fork of OSSEC by a community led by **Santiago Bassett**
- **Goal**: Extend and modernize OSSEC features, address stagnation, improve scalability
- **Fully open-source**, under GPL v2

💡 **Key Improvements over OSSEC:**

Feature	OSSEC	Wazuh
Elastic Stack integration	Manual, limited	Native integration with ELK
RESTful API	None	Full-featured API
Scalability	Moderate	High (multi-node architecture)
Active development	Slowed post-2008	Ongoing, active community & team
Modules & Threat Intel	Basic	Advanced rules, MITRE ATT&CK map
UI Dashboard	Basic (or 3rd-party)	Built-in Kibana app

📅 **Wazuh Milestones:**

Year	Milestone
2015	Wazuh forked from OSSEC
2017	Released Wazuh 2.0 with full ELK integration
2019	Released Wazuh 3.x (improved cluster features)
2020	Added MITRE ATT&CK mapping, Cloud monitoring modules
2022	Wazuh 4.x: complete platform with Security Analytics, XDR
2024	Continues as full-fledged **Security Platform**

🔁 From OSSEC to Wazuh: The Transition Explained

1. Background: OSSEC's Origins & Limitations

Feature	Description
Tool	OSSEC (Open Source Security Event Correlator)
Created	2004 by Daniel B. Cid
Acquired by	Trend Micro in 2008
Purpose	Lightweight, open-source Host-based Intrusion Detection System (HIDS)
Strengths	Log monitoring, File Integrity Monitoring (FIM), Active response
Limitations	Slower development, no REST API, no native SIEM/Elastic integration, limited scalability

🚨 Key Limitation:

- After the Trend Micro acquisition, OSSEC updates slowed, community involvement declined, and modern features were lacking (e.g., cloud support, dashboards, scalability).

2. Birth of Wazuh (2015)

Aspect	Detail
Forked From	OSSEC source code
👨 Founded By	Santiago Bassett
Mission	Modernize OSSEC, make it scalable, extensible, and integrate natively with SIEMs
Initial Focus	Keep OSSEC's HIDS core but build new architecture and features around it

3. Purpose of Wazuh – Why Was It Created?

Problem with OSSEC	Wazuh's Solution
Outdated UI	Modern Kibana-based dashboard
No cloud monitoring support	AWS, Azure, GCP modules
No MITRE ATT&CK support	Built-in ATT&CK matrix mapping
No native SIEM integration	Native Elastic Stack integration
Limited scalability (single-node)	Clustered manager & multi-node support
No API	Full-featured RESTful API
Fewer log types supported	Custom decoders, more OS/app support

🛠️ 4. Major Enhancements in Wazuh vs OSSEC

Feature	OSSEC	Wazuh
Active Development	Slowed	Ongoing & fast-paced
Web UI	Minimal	Built-in Wazuh App for Kibana
Agent Scalability	Limited	Cluster-ready
REST API	None	Full REST API
Threat Intelligence	Basic	Supports integration with OTX, etc.
SIEM Integration	Manual	Native Elastic Stack support
MITRE ATT&CK	No	Yes
Cloud Monitoring	No	AWS, Azure, GCP support
Compliance Frameworks	No	PCI DSS, GDPR, NIST, HIPAA, etc.

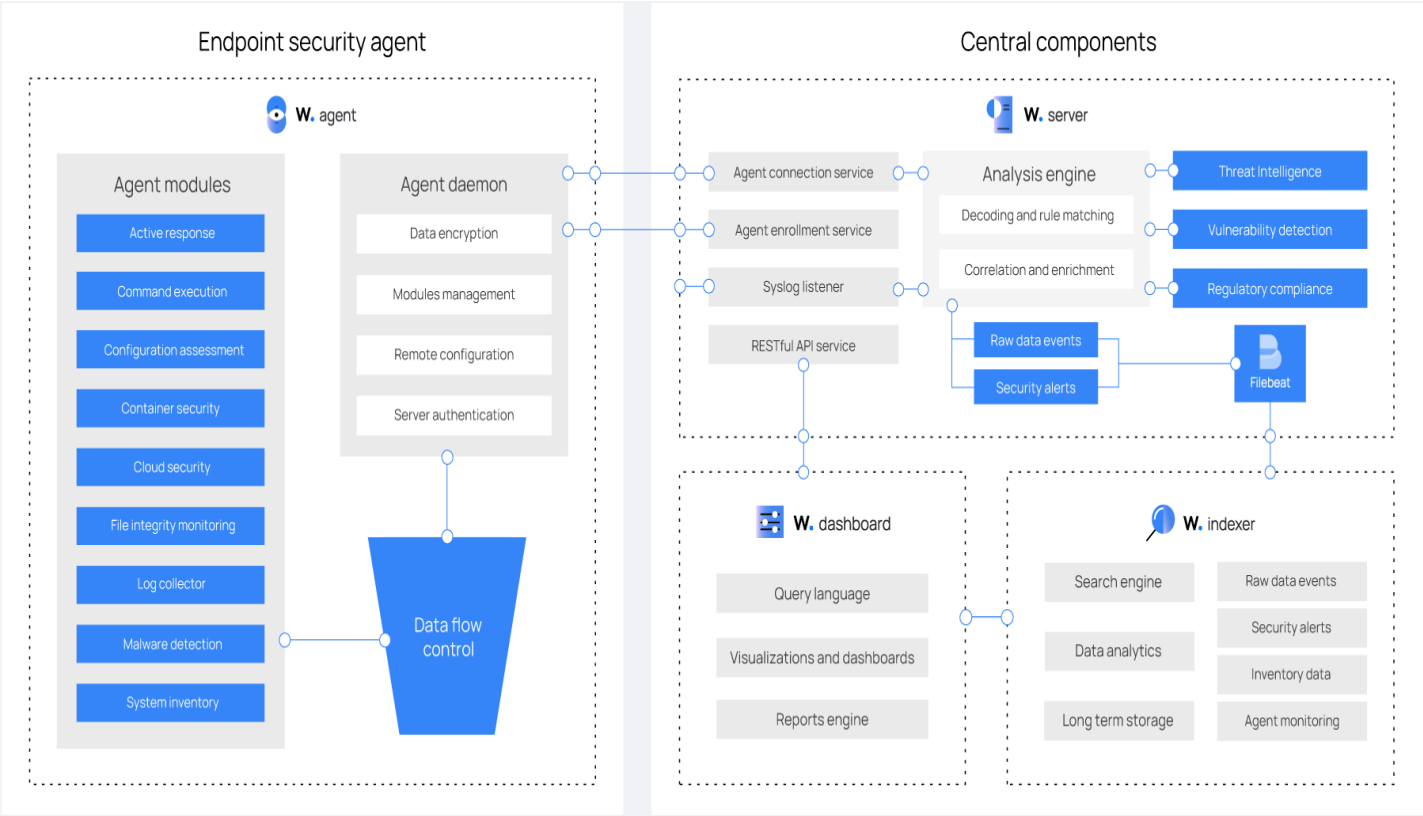
🧭 5. Timeline Snapshot

Year	Milestone
2004	OSSEC launched by Daniel B. Cid
2008	Trend Micro acquires OSSEC
2015	Wazuh forked from OSSEC
2017	Wazuh 2.0 – Elastic Stack integration
2019	Wazuh 3.x – Clustering & API enhancements
2022	Wazuh 4.x – Full XDR platform capabilities

6. Key Takeaways

- Wazuh is not just an OSSEC replacement; it's a full security platform.
- OSSEC served as the core engine, but Wazuh built an entire modern ecosystem around it.
- Wazuh is suited for enterprise environments, hybrid cloud, and regulatory compliance.
- Wazuh retains agent-based HIDS principles while adding SIEM, EDR, and XDR capabilities.

Overview of Wazuh Components



🛡️ Wazuh Endpoint Security Agent

The Wazuh agent is a lightweight software installed on monitored endpoints (e.g., Linux, Windows, macOS). It collects system data and security events, processes them locally, and securely sends results to the Wazuh Manager.

1. Major Agent Components

A. Agent Module (Functional Layer)

These modules perform security tasks on the endpoint.

Module	Purpose
Active Response	Automatically reacts to threats (e.g., block IP, kill process) using predefined rules.
Command Execution	Executes commands/scripts on the agent for checks or automated responses.
Configuration Assessment	Audits system configurations for compliance (e.g., CIS, NIST rules).
Container Security	Monitors Docker containers for vulnerabilities, runtime behavior, etc.

Module	Purpose
Cloud Security	Collects cloud metadata and events (for cloud-hosted agents).
File Integrity Monitoring (FIM)	Tracks file changes (create/modify/delete) on important system files.
Log Collection	Collects and parses logs (e.g., syslog, application, Windows Event Logs).
Malware Detection	Detects malicious files/processes using YARA rules, VirusTotal, etc.
System Inventory	Collects hardware/software asset data (packages, versions, users).

These modules are plug-and-play — enabled/disabled via configuration.
Results are sent to the Agent Daemon for processing and transmission.

B. Agent Daemon (wazuh-agentd) (Control Layer)

Handles communication, security, and configuration management.

Component	Role
Data Encryption	Encrypts data sent to the manager (typically using AES).
Modules Management	Starts/stops internal agent modules as needed.
Remote Configuration	Allows the Wazuh manager to remotely modify agent configs.
Server Authentication	Validates identity of the manager before accepting configs or commands.

2. Wazuh Agent Dataflow (End-to-End)

Here's how data moves from your endpoint to the Wazuh manager:

📌 Step-by-Step Flow:

```
[ Files / Logs / Configs / Processes ]
          ↓
   🔍 Agent Modules (FIM, Logs, Malware, etc.)
          ↓
    📦 Preprocessed data/events
          ↓
      🧠 Agent Daemon (wazuh-agentd)
        ├── Encrypts data
        ├── Validates server
        ├── Applies remote config
```

↓
🔐 Encrypted & Authenticated Communication
↓
📬 Wazuh Manager (receiver service)

Response Flow (Manager → Agent)

If Wazuh manager wants to send instructions back to agent:

[Manager Rules/Policies/Commands]
↓
🛡️ Encrypted instructions (via agent-auth / API)
↓
🧠 Agent Daemon receives & validates
↓
🎯 Executes command / updates config / triggers response
↓
📁 Result logged & reported back to manager

3. Quick Recap: Functional vs Control Layer

Layer	Component	Key Role
Functional Layer	Agent Module	Security tasks (FIM, log analysis, malware detection, etc.)
Control Layer	Agent Daemon	Manages modules, handles encryption, config, and server trust

Wazuh Central Components

Wazuh Server

The Wazuh Server is the core processing and analysis unit. It receives data from agents, processes and enriches it, and then forwards structured alerts to the indexer (Elastic/OpenSearch).

Key Functions of Wazuh Server:

Sub-Component	Description
Agent Connection State	Manages persistent connections with agents (via wazuh-remoted); monitors status (active, disconnected).
Agent Enrollment Service	Handles registration/authentication of new agents. Supports manual or automatic registration.
Syslog Listener	Accepts syslog-formatted logs from external systems (firewalls, routers, etc.).

Sub-Component	Description
RESTful API Service	Offers programmatic access for querying data, managing agents, and more. Used by the Dashboard.
Analysis Engine	Core engine that processes incoming data with decoders and rules.
Decoding / Rule Matching	Converts raw logs into structured data (decoders), then applies detection logic (rules).
Correlation & Enrichment	Adds context using threat intel, MITRE ATT&CK mapping, GeoIP, etc.
Security Alert Generation	Creates alerts based on matching rules, thresholds, etc.
Threat Intelligence	Compares data against threat intel feeds like OTX, VirusTotal.
Vulnerability Detection	Integrates with CVE databases and Wazuh's vulnerability detector to flag outdated or vulnerable software.
Regulatory Compliance	Checks against policies like PCI DSS, HIPAA, NIST, etc.

🧭 Wazuh Dashboard & Indexer

1. Wazuh Dashboard

The Wazuh Dashboard is a customized Kibana-based web interface that provides real-time visibility into all data processed by the Wazuh platform.

🔍 A. Key Features

Feature	Description
Visualizations & Dashboards	Custom charts, graphs, maps for threat levels, agent status, FIM events, vulnerabilities, etc.
Query Language Support	Uses Lucene Query Language or OpenSearch Query DSL for advanced filtering and searches.
Report Engine	Generates scheduled or on-demand PDF/HTML reports with visual data from dashboards.
Dashboard Customization	Users can build their own dashboards for specific compliance views, threats, or business use cases.
Live Agent Monitoring	Displays current agent status (active/disconnected), OS info, alerts in real time.
Pre-built Panels	Comes with built-in panels for MITRE ATT&CK, compliance, vulnerability detection, FIM, etc.

🧪 B. Query Language Examples

Use Case	Example Query (Lucene)
Get high severity alerts	rule.level: >=10
Search for SSH login attempts	data.fields.system.auth.program: "sshd"
Filter by agent name	agent.name: "webserver01"
FIM file change events	rule.groups: "syscheck"
CVE alerts	vulnerability.cve: "CVE-2024-23897"

🔎 2. Wazuh Indexer (OpenSearch / Elasticsearch)

Acts as the search engine, analytics engine, and long-term storage layer for all Wazuh alert and inventory data.

🔍 A. Key Responsibilities

Role	Description
Search Engine	Enables high-speed full-text and fielded search across all data
Data Analytics	Supports aggregations, statistics, filtering (used for dashboards and reports)
Long-Term Storage	Stores all structured security alerts, raw events, and metadata with time-based indices
Raw Data Events	Stores original log content before processing, if configured (e.g., syslog, Windows Event Log)
Security Alerts	Stores enriched and correlated alerts from the Wazuh server
Inventory Data	Stores agent-collected asset data: installed apps, running processes, hardware info
Agent Monitoring	Tracks heartbeat, last check-in time, agent status metadata
Index Lifecycle Management (ILM)	Controls index rotation, compression, and retention policies

B. Data Types Stored

Data Type	Source Module
Security Alerts	Rule engine (Wazuh server)
Raw Log Data	Log collection, syslog, FIM
Vulnerabilities	Vulnerability Detector
Inventory Info	System Inventory module
Agent Metadata	Agent daemon heartbeat
Compliance Results	Configuration Assessment

Full Data Lifecycle (Platform Perspective)

```
[ Agent Modules ]
    ↓
[ Agent Daemon ]
    ↓
🔐 Encrypted Event Data
    ↓
[ Wazuh Server ]
    ├─ Decoders & Rule Matching
    ├─ Threat Intelligence & MITRE ATT&CK
    └─ JSON-formatted Alerts
    ↓
[ Wazuh Indexer ]
    ├─ Indexing (Searchable Format)
    ├─ Stored in Time-based Indices
    └─ Queried by Dashboard
    ↓
[ Wazuh Dashboard ]
    ├─ Custom Dashboards
    ├─ Reports
    └─ Real-Time Alert Views
```

4. Filebeat (Optional/Used in Specific Setups)

Role	Description
Log Shipper	Sends logs (from Wazuh logs, JSON alerts, or syslog) to external indexers or SIEMs.
Use Case	Needed when Wazuh sends raw logs to separate Elasticsearch/OpenSearch cluster.
Forwarder Agent	Supports buffering, load balancing, and secure delivery.