PROJECT

Multi-tenant cybersecurity operations centre designed as an open cloud service with machine learning elements

Supporting document for V4

Overview of SOC components

Linked to work package KPB2 – Design of the SOCaaS service

Funded by the European Union NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1. Categories of tools in the SOC centre and examples 4
1.1. Log and data collection tools (Log Collectors / Agents) 4
1.2. Log processing and forwarding tools (Log Processors / Forwarders) 4
1.3. Log storage and indexing tools (Data Storage / Indexing) 4
1.4. Attack analysis and threat detection tools (Threat Detection/IDS /SIEM) 4
1.5. Incident management and response tools (Incident Response /Automation) 4
1.6. Visualization and monitoring tools (Dashboards / Monitoring Tools) 5
1.7. Container and cloud security monitoring tools (Cloud /Container Security) 5
1.8. Supplementary and auxiliary tools (Utility Tools) 5
2. Detailed analysis of SOC tools 5
2.1. Log and data collection tools (Log Collectors / Agents) 5
2.1.1. Filebeat 5
2.1.2. Metricbeat 6
2.1.3. Packetbeat 6
2.1.4. Syslog-ng 6
2.1.5. Auditbeat 7
2.2. Log processing and forwarding tools (Log Processors / Forwarders) 7
2.2.1. Logstash 7
2.2.2. Fluentd 8
2.2.3. Vector (Vector.dev) 8
2.3. Log storage and indexing tools (Data Storage / Indexing) 9
2.3.1. Elasticsearch 9
2.3.2. OpenSearch 9
2.3.3. TimescaleDB 10
2.3.4. InfluxDB 10
2.4. Attack analysis and threat detection tools (Threat Detection/IDS /SIEM) 10
2.4.1. Suricata 10
2.4.2. Snort 11
2.4.3. Wazuh 11
2.4.4. OSSEC 12
2.4.5. Zeek (formerly Bro) 12
2.5. Incident management and response tools (Incident Response /Automation) 13
2.5.1. TheHive 13
2.5.2. Cortex 13
2.5.3. MISP 14

2.6. Visualization and monitoring tools (Dashboards / Monitoring Tools) 14
2.6.1. Kibana 14
2.6.2. Grafana 15
2.6.3. Prometheus 15
2.6.4. Loki 15
2.7. Container and cloud security monitoring tools 16
2.7.1. Falco 16
2.7.2. Trivy 16
2.7.3. Kube-hunter 17
2.8. Supplementary and auxiliary tools (Utility tools) 17
2.8.1. Nmap 17
2.8.2. OpenVAS 18
2.8.3. Zabbix / Libre NMS 18
2.8.4. Ansible 18
3. Wazuh 19
3.1. System architecture 19
3.2. Installation requirements 19
3.2.1. Wazuh Indexer 19
3.2.2. Wazuh Server 20
3.2.3. Wazuh Dashboard 20
3.3. Supported systems and predefined decoders in Wazuh 20
3.3.1. Operating systems and platforms 20
3.3.2. Security solutions and antivirus software 21
3.3.3. Firewalls, IDS/IPS and network devices 21
3.3.4. Authentication and access systems 21
3.3.5. E-mail servers and services 21
3.3.6. Web servers, applications and CMS 22
3.3.7. Virtualization and containers 22
3.3.8. Cloud and services 22
3.3.9. Database systems 22
3.3.10. Other services, logs and tools 22
List of sources 23

1. CATEGORIES OF TOOLS IN THE SOC CENTRE AND EXAMPLES

1.1. Log and data collection tools (Log Collectors / Agents)

They collect logs, metrics and network traffic from servers, client devices and network elements.

- **Filebeat** – collection of system and application logs
- **Metricbeat** – collection of metric data (CPU, memory, disk...)
- **Packetbeat** – monitoring of network communication
- **Syslog-ng** – classic log receiver for Unix systems
- **Auditbeat** – monitoring of system events (e.g. user login)

1.2. Log processing and forwarding tools (Log Processors / Forwarders)

They transform, filter or route logs to the target storage.

- **Logstash** – centralized log processor (parsing, enrichment)
- **Fluentd** – an alternative to Logstash, lightweight and flexible
- **Vector.dev** – modern high-performance log forwarder

1.3. Log storage and indexing tools (Data Storage / Indexing)

They store logs and enable them to be searched, filtered and analysed quickly.

- **Elasticsearch** – distributed full-text search engine and log database
- **OpenSearch** – a fork of Elasticsearch (AWS), suitable for open-source solutions
- **TimescaleDB / InfluxDB** – for time-series metrics, mainly in monitoring

1.4. Attack analysis and threat detection tools (Threat Detection / IDS / SIEM)

They are used to detect suspicious or malicious activities.

- **Suricata** – network IDS/IPS (packet analysis)
- **Snort** – alternative IDS tool
- **Wazuh** – host-based incident detection, file integrity monitoring
- **OSSEC** – simpler HIDS tool (older version of Wazuh)
- **Zeek (Bro)** – higher-level network analysis (forensic analysis)

1.5. Incident management and response tools (Incident Response / Automation)

They make it possible to record incidents, assign tasks and respond to threats.

- **TheHive** – web platform for recording and handling incidents
- **Cortex** – tool for automated responses, connected to TheHive

- **MISP** – sharing of threats and IoCs (Indicators of Compromise) with other teams

1.6. Visualization and monitoring tools (Dashboards / Monitoring Tools)

They are used to create clear dashboards, graphs and alerts.

- **Kibana** – primary tool for visualizing data from Elasticsearch
- **Grafana** – advanced visualization tool for metrics from Prometheus, Elastic, etc.
- **Prometheus** – tool for collecting and storing time-series metrics (e.g. CPU, RAM, network)
- **Loki** – a tool from Grafana for log visualization

1.7. Container and cloud security monitoring tools (Cloud / Container Security)

They detect suspicious activities in containers, the cloud or the CI/CD pipeline.

- **Falco** – runtime security tool for Kubernetes / Docker
- **Trivy** – vulnerability scanner for containers and Docker images
- **Kube-hunter** – detection of weaknesses in a Kubernetes cluster

1.8. Supplementary and auxiliary tools (Utility Tools)

Tools for network analysis, scanning and administration.

- **Nmap** – port and network scanner (used in analysis)
- **OpenVAS** – tool for finding vulnerabilities in systems
- **Zabbix / LibreNMS** – monitoring of device status (ping, availability)
- **Ansible** – automation of the deployment and management of tools

2. DETAILED ANALYSIS OF SOC TOOLS

2.1. Log and data collection tools (Log Collectors / Agents)

2.1.1. Filebeat

Description:

Filebeat is a lightweight agent developed by Elastic, designed to collect and send logs from various systems (e.g. system logs, application logs, web server logs, database logs, etc.) to a processing tool such as Logstash or Elasticsearch. It runs on the client device as a daemon, continuously reads log files (e.g. /var/log/syslog) and forwards them.

Advantages:

- Very low consumption of system resources.
- Simple installation and configuration.
- Reliable log transfer thanks to a built-in buffer.
- Support for various log types via modules (e.g. Apache, Nginx, MySQL).
- Possibility of direct connection to Elasticsearch.

Disadvantages:

- Limited options for log processing or transformation.
- With large volumes of logs it may be less powerful than other solutions
- A higher level of security (TLS, authentication) requires additional configuration

2.1.2. Metricbeat

Description:

Metricbeat is an agent for collecting system and application metrics. It gathers data such as CPU, memory, disk and network traffic usage, as well as statistics from various applications (e.g. Docker, MySQL, Apache). It sends the data to Elasticsearch or via Logstash.

Advantages:

- A simple way of monitoring the performance of servers and services.
- Broad module support for well-known applications and systems.
- Low consumption of system resources.
- Suitable for integration with Grafana or Kibana for visualization.

Disadvantages:

- Less flexible compared with tools such as Prometheus.
- Some metrics may be missing for specific applications.
- Requires correct setting of intervals, otherwise it may flood the output or the system.

2.1.3. Packetbeat

Description:

Packetbeat is a tool for passive monitoring of network communication. It captures packets directly from the network interface and analyses them according to protocol (e.g. HTTP, DNS, MySQL, PostgreSQL, Redis, etc.). It sends the outputs to Elasticsearch.

Advantages:

- It enables detailed analysis of network traffic without the need to modify applications.
- Suitable for detecting non-standard or suspicious network behaviour.
- Simple integration into the Elastic Stack.

Disadvantages:

- Higher demands on network and computing resources with a large number of packets.
- Limited range of supported protocols.
- Less suitable for encrypted transmission (HTTPS, TLS).

2.1.4. Syslog-ng

Description:

Syslog-ng is an advanced logging daemon that receives, processes and distributes logs from systems and applications. It functions as a central logging point for Unix/Linux systems. It can receive logs via protocols such as UDP, TCP and TLS.

Advantages:

- Suitable as a central logging gateway for the entire network.
- High configurability and support for log filtering.
- Support for encrypted log transfer.
- It can write logs to files or databases, or forward them to tools such as Logstash.

Disadvantages:

- More complicated configuration in complex scenarios.
- It does not include its own analytical or visualization functions.
- It requires an additional system for further processing or storage of logs (e.g. Elasticsearch).

2.1.5. Auditbeat

Description:

Auditbeat is an agent from the Elastic Beats package that gathers information about events at the operating system level. It monitors, for example, login attempts, process launches and changes in system files, and can be connected to the Linux Audit Framework.

Advantages:

- It enables the detection of security incidents at the operating system level.
- It monitors critical system activities (e.g. changes in /etc/passwd, execution of binaries).
- Excellent integration with the Elastic Stack.

Disadvantages:

- Greater difficulty in setting audit rules correctly.
- With a large number of events it can generate large volumes of data.
- It requires advanced knowledge of the operating system for effective use.

2.2. Log processing and forwarding tools (Log Processors / Forwarders)

2.2.1. Logstash

Description:

Logstash is a powerful and widely used tool from Elastic for processing, filtering, transforming and routing logs and other types of data. It works as a "pipeline": it receives input data, processes it according to defined rules (filters, regexes, geoIP, JSON parsing...) and then sends it to the output system, most often to Elasticsearch.

Advantages:

- Strong integration with the Elastic Stack.

- Large number of available plugins (input, output and filter plugins).
- Capability for complex data processing.
- Proven solution with a large community and documentation.
-

Disadvantages:

- Higher demands on computing resources (memory, CPU).
- More complicated configuration (e.g. compared with Fluentd).
- It may have lower performance with a large number of parallel inputs without optimization.

2.2.2. Fluentd

Description:

Fluentd is an open-source data collector designed to unify log processing and route logs to various systems (Elasticsearch, Kafka, InfluxDB, etc.). It is an alternative to Logstash with an emphasis on low resource consumption and modularity. Its configuration is based on a simple structure (input → filter → output rules).

Advantages:

- Very lightweight and efficient (lower demands than Logstash).
- Huge community and plugin ecosystem.
- Flexible and extensible (plugins in Ruby and C).
- It can also work as a log aggregator between multiple machines.

Disadvantages:

- It requires knowledge of the structure for advanced settings.
- Plugins vary in quality (not all are officially supported).
- Slightly more complicated debugging in the event of configuration errors.

2.2.3. Vector (Vector.dev)

Description:

Vector is a modern tool for collecting, processing and forwarding logs or metrics, developed by Datadog (originally Timber.io). It is designed for high performance, low latency and secure data delivery. It can function both as an agent and as a central forwarder, and supports inputs such as syslog, journald and Prometheus, as well as outputs such as Elasticsearch, Kafka and ClickHouse.

Advantages:

- Written in Rust – extremely powerful and secure.
- Simple, readable configuration (TOML or JSON).
- Low memory load, suitable for large volumes of data.
- Strong focus on observability and reliability (including retry logic and buffering).

Disadvantages:

- Smaller community compared with Logstash or Fluentd.
- Some integrations and plugins are not yet as widespread (a newer project).

- It requires learning a new syntax and concepts for those who are used to Logstash.

2.3. Log storage and indexing tools (Data Storage / Indexing)

2.3.1. Elasticsearch

Description:

Elasticsearch is a distributed search and analytics engine that functions as a database for storing, indexing and searching logs, text data and metric data. It is a core component of the Elastic Stack. It is used for fast searching and filtering of large amounts of data, for example system logs, network events or incidents.

Advantages:

- Very fast and powerful search over huge volumes of data.
- Support for both full-text and structured search.
- Distributed architecture – enables scaling and high availability.
- Natively integrated with Logstash, Beats and Kibana (Elastic Stack).
- Strong aggregation and analytics capabilities over data.

Disadvantages:

- Higher hardware demands with larger data volumes.
- With incorrect configuration it can quickly consume memory and CPU.
- Since version 7.11, core functions have moved under a commercial licence (Elastic License), which may be restrictive for some open-source projects.

2.3.2. OpenSearch

Description:

OpenSearch is an open-source fork of Elasticsearch and Kibana, developed under the auspices of Amazon Web Services (AWS). It was created in response to the change in licensing terms in the Elastic Stack. It offers functionality similar to Elasticsearch, including full-text search, indexing of logs and data, analytics and visualization.

Advantages:

- It retains the core features of Elasticsearch but remains fully open-source.
- Regularly developed by the community and the AWS team.
- Compatible with existing solutions built on Elasticsearch 7.x.
- It includes its own visualization tool – OpenSearch Dashboards.

Disadvantages:

- It is not 100% backward compatible with newer versions of Elasticsearch.
- Smaller community and documentation compared with the original Elastic Stack.
- Some advanced features of the Elastic Stack (e.g. Machine Learning) are not available or are less mature.

2.3.3. TimescaleDB

Description:

TimescaleDB is an extension of the PostgreSQL database, optimized for time-series data. It is used mainly for storing and analysing metric data from monitoring systems. It has the advantages of a classic relational database, while also offering excellent performance when working with time as a dimension.

Advantages:

- Built on PostgreSQL – excellent familiarity and stability.
- Great for metrics from monitoring tools (e.g. from Grafana).
- Excellent performance when working with historical data.
- Advanced analytical functions over time-series data.

Disadvantages:

- It is not intended for text logs or full-text search.
- Higher complexity when scaling in very large environments.
- It does not include "elastic-style" indexing and search.

2.3.4. InfluxDB

Description:

InfluxDB is a specialized database for storing time-series data (time-series database). It is widely used for monitoring the performance of servers, networks, IoT devices, etc. Data are stored in time series and aggregated very efficiently.

Advantages:

- Extremely fast at writing and reading time-series data.
- An ideal solution for systems such as Telegraf, Grafana and Kapacitor.
- Good for both short-term and long-term analysis of metric data.
- Simple installation and integration with monitoring systems.

Disadvantages:

- It is not intended for processing classic text logs.
- Some features in InfluxDB 2.x are available only in the licensed version.
- More complicated replication and scaling with large clusters.

2.4. Attack analysis and threat detection tools (Threat Detection / IDS / SIEM)

2.4.1. Suricata

Description:

Suricata is an open-source network IDS/IPS and packet analyser that can analyse network communication in real time, search for known attacks using signatures and generate alerts. In addition to threat detection, it also supports file transfer functionality, protocol analysis (HTTP, TLS, DNS...) and performance metrics.

Advantages:

- Advanced real-time packet and protocol analysis.
- Support for multi-threaded processing – high performance.
- Ability to function as both an IDS and an IPS.
- Extensible using rules (compatible with Snort signatures).
- Integrated with tools such as Wazuh or Elasticsearch.

Disadvantages:

- Higher configuration complexity in larger networks.
- Under heavy load, the hardware must be carefully sized.
- It requires regular updating of signatures (e.g. from ET Open).

2.4.2. Snort

Description:

Snort is an open-source network intrusion detection tool (IDS) that inspects network traffic using signatures and heuristics. It is one of the oldest and most widely used IDS tools and is used mainly to detect known attacks and suspicious network behaviour.

Advantages:

- Large database of attack signatures.
- Broad community and many available configuration examples.
- Suitable for small to medium-sized networks.
- Can be used as both an IDS and an IPS.

Disadvantages:

- Single-threaded architecture – lower performance than Suricata with heavy traffic.
- Less extensive protocol analysis.
- Less active development compared with Suricata.

2.4.3. Wazuh

Description:

Wazuh is a host-based IDS (HIDS) that monitors activity directly on individual devices – e.g. system changes, login attempts and changes in files. It is a modern solution built on OSSEC and is closely integrated with the Elastic Stack. It also supports agents for Windows, Linux and macOS.

Advantages:

- Excellent integration with Elasticsearch and Kibana.
- It records changes in files, user actions, logins and suspicious activity.
- It works with a threat database (threat intelligence).
- Agent-based system – suitable for various operating systems.
- It also includes a compliance monitoring module (e.g. PCI-DSS, GDPR...).

Disadvantages:

- It requires management and updating of agents on all devices.
- It can generate a large amount of data (logs) if not set up correctly.
- Higher complexity in large-scale deployments without automation.

2.4.4. OSSEC

Description:

OSSEC is an open-source HIDS (host-based intrusion detection system) that monitors files, logs, user actions and the integrity of system files. It is the predecessor of Wazuh, which split off from it through further development. It is simpler and less integrated, but still effective in small environments.

Advantages:

- Low system requirements.
- Simple installation and configuration.
- Support for multi-platform agents (Windows, Linux, BSD).
- Suitable for smaller deployments without the need for a complex visual interface.

Disadvantages:

- It offers neither a native web interface nor integration with the Elastic Stack.
- Weaker community and limited development compared with Wazuh.
- Less suitable for large enterprise deployments.

2.4.5. Zeek (formerly Bro)

Description:

Zeek is an advanced network analysis tool that enables detailed monitoring of network traffic at a higher layer than Suricata or Snort. It does not work with signatures, but with its own scripting language, which can be used to define detection rules and scenarios. It is used mainly for forensic analysis.

Advantages:

- Highly configurable and scriptable (its own language).
- It enables complex analysis of network behaviour (behavioural models).
- Suitable for in-depth forensic analysis and long-term trend monitoring.
- It can export structured logs for further processing.

Disadvantages:

- It requires experience with Zeek scripting.
- It is not intended for signature-based detection like Snort or Suricata.
- It may be more performance-intensive with extensive network traffic.

2.5. Incident management and response tools (Incident Response / Automation)

2.5.1. TheHive

Description:

TheHive is an open-source security incident management platform (SIRP – Security Incident Response Platform). It is used for recording incidents, classifying them, distributing tasks within the team and tracking their handling. Each incident (a so-called case) can contain multiple alerts, artefacts (e.g. IP addresses, hashes), comments and response steps. The platform has a web interface and a REST API, which enables easy integration with other systems such as Cortex and MISP.

Advantages:

- Clear recording of incidents and workflow.
- Support for team collaboration (task assignment, comments).
- Flexible incident templates (automatic filling of fields).
- Strong integration with Cortex and MISP.
- Open-source with an active community.

Disadvantages:

- It requires technical knowledge for deployment and administration.
- The interface may not be intuitive for less experienced users.
- Occasional need for updates/patches, which may break the compatibility of add-ons.
- Some functions (e.g. reporting) may be limited without custom programming.

2.5.2. Cortex

Description:

Cortex is an open-source tool for automating incident analysis and response. It is designed as an extension of TheHive, but can also function on its own. It uses so-called analysers (e.g. WHOIS lookup, VirusTotal scan, IP geolocation) and responders (e.g. blocking an IP in the firewall, deactivating a user account). These modules are executed on the basis of artefacts from incidents and can be triggered manually or automatically.

Advantages

- It automates routine and repetitive activities.
- It enables rapid response to threats (playbooks).
- Modular architecture – custom analysers/responders are easy to add.
- Efficient integration with TheHive (in a single click).
- It extends the analytical capabilities of analysts.

Disadvantages

- It requires configuration and the setting up of authentication (API keys, tokens).
- With a large number of requests, some analysers may cause delays.
- Not all analysers/responders are regularly updated.
- Some modules may depend on paid services

2.5.3. MISP

Description

MISP is an open-source platform for sharing threat information (Threat Intelligence). It makes it possible to create and manage a database of so-called events, which contain IoCs (Indicators of Compromise), such as malicious IP addresses, file hashes, domains, e-mails and other threat-related data. MISP supports standards such as STIX/TAXII and enables automatic synchronization with other MISP instances, thereby supporting community information sharing.

Advantages

- It supports collaboration between teams and organizations.
- Possibility of creating custom TI databases.
- Compatibility with Threat Intelligence standards (TAXII/STIX).
- Powerful API for integration with other tools (e.g. SIEM, TheHive).
- It allows categories and taxonomies to be customized.

Disadvantages

- The user interface may be complicated for beginners.
- It requires knowledge of working with IoCs and taxonomies.
- Synchronization with other MISP instances can be technically demanding.
- Need for regular database maintenance (cleaning up outdated events).

2.6. Visualization and monitoring tools (Dashboards / Monitoring Tools)

2.6.1. Kibana

Kibana is a visualization tool developed by Elastic, used primarily for visualizing and analysing data stored in Elasticsearch. It makes it possible to create clear dashboards, graphs, boards and timelines, and to search logs. It is often part of the ELK stack (Elasticsearch, Logstash, Kibana), which is used for processing and visualizing logs.

Advantages

- Strong integration with Elasticsearch.
- Interactive and flexible visualizations.
- Fast search in huge amounts of data (logs, IoCs, metrics).
- Suitable for analysing both security and performance data.

Disadvantages

- It requires Elasticsearch as a backend (it cannot be used on its own).
- Some advanced features are available only in commercial versions.
- More complicated setup when integrating with multiple data sources.
- Less intuitive than Grafana for quickly creating dashboards.

2.6.2. Grafana

Grafana is an open-source visualization tool that makes it possible to display metrics from various sources – mainly from Prometheus, but also from Elastic, InfluxDB, MySQL and others. It is extremely flexible and suitable for creating overview dashboards for performance, infrastructure or security monitoring.

Advantages

- Support for many data sources (Prometheus, Loki, Elasticsearch...).
- Very modern and visually attractive dashboards.
- Possibility of setting up alerts and panels, and sharing dashboards via links.
- Strong community and many ready-made templates.

Disadvantages

- It requires a connection to an external data source – it does not collect data itself.
- Setting up alerting may be complicated for beginners.
- More advanced features (e.g. reporting, access roles) are available in paid editions.

2.6.3. Prometheus

Description

Prometheus is an open-source system for collecting, storing and processing time-series metrics. It is designed mainly for monitoring infrastructure and applications. It regularly queries (scrapes) metrics from various sources (exporters) and stores them in its own TSDB (time-series database). It is most often used in combination with Grafana.

Advantages

- Strong support for **time-series metrics** (CPU, RAM, network, applications).
- Its own query language, **PromQL**, for data analysis.
- Large number of available **exporters** (e.g. for Linux, Docker, PostgreSQL...).
- Possibility of setting up alerts using Alertmanager.

Disadvantages

- It does not collect logs (only metrics).
- The system is not ideal for long-term storage of large data (without extensions).
- It requires knowledge of PromQL, which has a steeper learning curve.
- Less suitable for processing unstructured data.

2.6.4. Loki

Description

Loki is an open-source system for collecting and searching logs, developed by the same team as Grafana. It is conceived as "Prometheus for logs", i.e. it retains the same philosophy: easy deployment, low requirements, tagging instead of indexing. It is used together with Grafana to visualize logs and create overview panels.

Advantages

- Simple integration with Grafana.
- Cheaper and faster than Elasticsearch with larger volumes of logs.

- It does not index entire logs – only metadata (significantly lower storage requirements).
- Possibility of using the same tools and syntax as Prometheus.

Disadvantages

- Limited search functionality compared with Elasticsearch.
- A younger project – fewer modules and integrations.
- It requires correct tagging for efficient searching.
- It works less well with highly unstructured logs.

2.7. Container and cloud security monitoring tools

2.7.1. Falco

Description

Falco is an open-source runtime security tool developed by the Sysdig project. It is designed to detect suspicious behaviour in container environments (Docker, Kubernetes). It monitors kernel system calls in real time and compares them with predefined security rules. It can detect, for example, a shell being launched in a container, writes to system directories, configuration changes, suspicious network traffic, etc.

Advantages

- **Runtime threat detection** – immediate capture of non-standard behaviour.
- Suitable for **both Kubernetes and Docker**.
- Rich library of rules + possibility of creating custom ones.
- It supports outputs to alerting tools (Slack, syslog, Prometheus...).
- Easy integration into the CI/CD pipeline.

Disadvantages

- Possible false positives – rule tuning is required.
- Higher performance load if many containers are monitored.
- It needs low-level access (to the kernel / system calls).
- Configuration may be more complicated for newcomers.

2.7.2. Trivy

Description

Trivy is a lightweight and fast open-source vulnerability scanner developed by Aqua Security. It is designed to analyse container images (Docker), Git repositories, Kubernetes manifests and CI/CD pipelines. It detects known vulnerabilities in libraries, OS packages or dependencies that are part of the image.

Advantages

- Simple to use.
- Fast scanning and detailed results.
- It also supports **Infrastructure as Code** (e.g. Terraform, Helm).
- It works locally, without the need for a complex installation.
- Regular updates of the vulnerability database (NVD, Red Hat, Debian...).

Disadvantages

- It does not scan runtime behaviour – only static images and configurations.

- It requires regular updating of the vulnerability database.
- With complex images it may generate many findings, not all of which are relevant (false positives).
- Some features are in the extended Aqua Enterprise version.

2.7.3. Kube-hunter

Description

Kube-hunter is an open-source tool for penetration testing of Kubernetes clusters, developed by Aqua Security. It is used to detect weaknesses in K8s configuration and deployment, such as open ports, APIs accessible without authentication, or missing restrictions on communication between pods.

Advantages

- Specialized in Kubernetes security.
- It supports both active and passive scanning modes.
- Comprehensible outputs – each finding has a description, a severity level and a recommendation.
- Suitable as a security check before deployment to production.
- It can be run as a container or a CLI tool.

Disadvantages

- Focused exclusively on **Kubernetes** – unusable outside K8s environments.
- Active tests may interfere with the running of applications (not recommended in production).
- It does not remove vulnerabilities, it only detects them.
- Less suitable for complex cloud environments outside K8s.

2.8. Supplementary and auxiliary tools (Utility tools)

2.8.1. Nmap

Description

Nmap (Network Mapper) is an open-source network scanner used for mapping networks and detecting open ports, services and operating systems on target devices. It is widely used in penetration tests, security audits and network diagnostics.

Advantages

- A very powerful and flexible tool.
- It supports scripting via the Nmap Scripting Engine (NSE).
- Fast scanning of large networks.
- Suitable for network administrators as well as ethical hackers.

Disadvantages

- It needs networking knowledge – it may be complex for beginners.
- Some scanning techniques may be perceived as "offensive".
- It does not include direct vulnerability outputs (for that, e.g. OpenVAS is used).

2.8.2. OpenVAS

Description

OpenVAS is an open-source tool for scanning vulnerabilities in systems, services and network devices. It is part of the Greenbone Vulnerability Management (GVM) package. It performs active scanning of targets and compares them against a database of known vulnerabilities (e.g. CVE).

Advantages

- Strong detection of known vulnerabilities.
- Possibility of regular automated checks.
- Usable for various types of systems (Windows, Linux, network devices).
- Graphical interface (GSA) as well as a CLI.

Disadvantages

- More complicated deployment and configuration.
- Higher hardware requirements.
- The scanner may generate many false positives.
- Longer time needed for a full scan.

2.8.3. Zabbix / Libre NMS

Description

Zabbix and LibreNMS are open-source tools for monitoring networks and devices. They make it possible to monitor the availability, latency and status of servers, switches, routers and other network elements using SNMP, ICMP, HTTP, agents and other protocols. They support alerting, metric visualization and automatic device discovery. Zabbix is a comprehensive system with high scalability and extensibility. LibreNMS is more focused on simple SNMP monitoring and quick installation.

Advantages

- Broad support for protocols and devices.
- Possibility of setting up notifications (e-mail, Telegram, Slack...).
- Suitable for continuous network monitoring.
- Open-source solutions without licence fees.

Disadvantages

- They require a dedicated server and configuration.
- Zabbix may be more complex for beginners.
- LibreNMS has a narrower range of functionality compared with Zabbix.
- Higher load in large installations.

2.8.4. Ansible

Description

Ansible is an open-source tool for automating IT processes, in particular software deployment, server configuration, updates and orchestration. It is agentless – it uses SSH and YAML playbooks to perform tasks on remote systems. It is widely used in DevOps, but also in security teams for the rapid deployment of rules, tools and configurations.

Advantages

- Simple and comprehensible syntax (YAML).
- It does not require an agent – it works via SSH.

- Suitable for both small and large infrastructures.
- It supports idempotency – executing it multiple times has the same result.

Disadvantages

- Less suitable for complex orchestration
- It requires SSH access and well-configured access permissions.
- More complicated debugging with large playbooks.

3. WAZUH

Wazuh is an open-source security platform focused on system integrity monitoring, threat detection, vulnerability management, log management and incident detection. Thanks to its modularity, it is suitable for use in both smaller and large IT environments.

Wazuh enables:

- Centralized collection and analysis of logs from servers and clients (Linux, Windows)
- Detection of suspicious behaviour based on rules and event correlation
- Monitoring of file changes, firewalls, antivirus software, authentication attempts and other security elements

3.1. System architecture

Wazuh consists of several components:

- **Wazuh server**: The central unit that receives data from agents, evaluates it according to rules and generates alerts.
- **Wazuh agent**: Lightweight software running on the monitored system (Linux, Windows) that collects logs, detects changes in the system and communicates with the server.
- **Wazuh indexer**: Stores and processes the collected data (logs, alerts, events).
- **Wazuh dashboard**: A graphical web interface for managing, visualizing and searching events.

3.2. Installation requirements

3.2.1. Wazuh Indexer

The Wazuh indexer requires a 64-bit Intel or AMD processor (x86_64/AMD64 architecture) and a Linux system in order to function.

Recommended OS

Amazon Linux 2, Amazon Linux 2023	**CentOS 7, 8**
Red Hat Enterprise Linux 7, 8, 9	Ubuntu 16.04, 18.04, 20.04, 22.04, 24.04

Hardware requirements

	Minimum		Recommended	
Component	**RAM (GB)**	**CPU (cores)**	**RAM (GB)**	**CPU (cores)**

Wazuh indexer	**4**	**2**	**16**	**8**

3.2.2. Wazuh Server

The Wazuh server requires a 64-bit Intel or AMD processor (x86_64/AMD64 architecture) and a Linux system in order to function.

Recommended OS

Amazon Linux 2, Amazon Linux 2023	**CentOS 7, 8**
Red Hat Enterprise Linux 7, 8, 9	Ubuntu 16.04, 18.04, 20.04, 22.04, 24.04

Hardware requirements

	Minimum		Recommended	
Component	**RAM (GB)**	**CPU (cores)**	**RAM (GB)**	**CPU (cores)**
Wazuh Server	**2**	**2**	**4**	**8**

3.2.3. Wazuh Dashboard

The Wazuh dashboard requires a 64-bit Intel or AMD processor (x86_64/AMD64 architecture) and a Linux system in order to function.

Recommended OS

Amazon Linux 2, Amazon Linux 2023	**CentOS 7, 8**
Red Hat Enterprise Linux 7, 8, 9	Ubuntu 16.04, 18.04, 20.04, 22.04, 24.04

Hardware requirements

	Minimum		Recommended	
Component	**RAM (GB)**	**CPU (cores)**	**RAM (GB)**	**CPU (cores)**
Wazuh indexer	**4**	**2**	**8**	**4**

3.3. Supported systems and predefined decoders in Wazuh

Wazuh contains a large number of predefined decoders and rules that enable automatic processing of logs from various operating systems, security solutions, network devices, databases and web applications. Thanks to this support, there is no need to manually create custom decoders for commonly used solutions. This chapter provides an overview of all the systems, vendors and services for which Wazuh has built-in support.

3.3.1. Operating systems and platforms

- Windows (windows_decoders.xml)
- macOS (macos_decoders.xml)
- OpenBSD (openbsd_decoders.xml)
- Solaris (solaris_decoders.xml)
- Unix/Linux in general (unix_decoders.xml)
- AIX (aix-ipsec_decoders.xml)

3.3.2. Security solutions and antivirus software

- Windows Defender (in windows_decoders.xml)
- ClamAV (clamav_decoders.xml)
- Kaspersky (kaspersky_decoders.xml)
- Sophos (sophos_decoders.xml, sophos_fw_decoders.xml)
- ESET Remote Administrator (eset-remote_decoders.xml)
- McAfee mcafee_decoders.xml)
- Trend Micro (OSCE) (trend-osce_decoders.xml)
- FireEye (fireeye_decoders.xml)
- Panda Security (PAPS) (panda-paps_decoders.xml)
- Symantec (symantec_decoders.xml)
- Cylance (cylance_decoders.xml)

3.3.3. Firewalls, IDS/IPS and network devices

- Cisco ASA, IOS, PIX, VPN, estreamer, FTD (cisco-*.xml)
- Fortinet – FortiGate, FortiDDoS, FortiMail, FortiAuth (forti*.xml)
- Palo Alto Networks (paloalto_decoders.xml)
- pfSense (pfsense_decoders.xml)
- Juniper – Junos (junos_decoders.xml)
- Barracuda (barracuda_decoders.xml)
- Huawei USG (huawei-usg_decoders.xml)
- SonicWall (sonicwall_decoders.xml)
- Check Point (checkpoint_decoders.xml, checkpoint-smart1_decoders.xml)
- Imperva (imperva_decoders.xml)
- Arbor Networks (arbor_decoders.xml)
- Suricata / Snort IDS (included in several rules such as snort_decoders.xml)

3.3.4. Authentication and access systems

- FreeIPA (freeipa_decoders.xml)
- RSA Authentication Manager (rsa-auth-manager_decoders.xml)
- Identity Guard (identity_guard_decoders.xml)
- OpenLDAP (openldap_decoders.xml)
- PAM (Pluggable Authentication Modules) (pam_decoders.xml)
- sudo / su / login commands (sudo_decoders.xml, su_decoders.xml)

3.3.5. E-mail servers and services

- Postfix (postfix_decoders.xml)
- Sendmail (sendmail_decoders.xml)
- Exim (exim_decoders.xml)
- Courier (courier_decoders.xml)
- Dovecot (dovecot_decoders.xml)
- Microsoft Exchange (MSEXchange) (msexchange-log-decoders.xml)
- FortiMail (fortimail_decoders.xml)

3.3.6. Web servers, applications and CMS

- Apache (apache_decoders.xml)
- Nginx (nginx_decoders.xml)
- WordPress (wordpress_decoders.xml)
- OwnCloud / Nextcloud (owncloud_decoders.xml, nextcloud_decoders.xml)
- GitLab (gitlab_decoders.xml)
- Jenkins (jenkins_decoders.xml)
- Horde Groupware (horde_decoders.xml)

3.3.7. Virtualization and containers

- VMware (vmware_decoders.xml)
- Docker (docker_decoders.xml)
- Proxmox VE (proxmox-ve_decoders.xml)
- FreePBS (freepbs_decoders.xml)

3.3.8. Cloud and services

- Microsoft Azure (azure_decoders.xml)
- AWS EKS Authenticator (aws-eks-authenticator_decoders.xml)
- QualysGuard (qualysguard_decoders.xml)

3.3.9. Database systems

- MySQL / MariaDB (mysql_decoders.xml, mariadb_decoders.xml)
- PostgreSQL (postgresql_decoders.xml)
- OracleDB (oracledb_decoders.xml)
- Redis (redis_decoders.xml)
- MongoDB (mongodb_decoders.xml)
- SQL Server (sqlserver_decoders.xml)
- TimescaleDB (usually via PostgreSQL)

3.3.10. Other services, logs and tools

- SSH / Telnet (ssh_decoders.xml, telnet_decoders.xml)
- FTP servers – vsftpd, proftpd, pure-ftpd, ftpd (several *_decoders.xml)
- ARPWatch, NTPD, Samba, racoon, dpkg, oscap, ossec, auditd – various system services
- Roundcube, Puppet, Suhosin, vshell, perdition, portsentry – specialized services and modules

LIST OF SOURCES

Elastic Beats

- Filebeat: https://www.elastic.co/beats/filebeat
- Metricbeat: https://www.elastic.co/beats/metricbeat
- Packetbeat: https://www.elastic.co/beats/packetbeat
- Auditbeat: https://www.elastic.co/beats/auditbeat
- Documentation: https://www.elastic.co/guide/en/beats/filebeat/current/index.html

Syslog-ng

- Documentation: https://www.syslog-ng.com/technical-documents/list/syslog-ng-open-source-edition
- Product page: https://www.syslog-ng.com/products/open-source-log-management/

Logstash

- Web: https://www.elastic.co/logstash
- Documentation: https://www.elastic.co/guide/en/logstash/current/index.html

Fluentd

- Web: https://www.fluentd.org/
- Documentation: https://docs.fluentd.org/

Vector.dev

- Web: https://vector.dev/
- Documentation: https://vector.dev/docs/

Elasticsearch

- Web: https://www.elastic.co/elasticsearch
- Documentation: https://www.elastic.co/guide/en/elasticsearch/reference/index.html

OpenSearch

- Web: https://opensearch.org/
- Documentation: https://opensearch.org/docs/latest/

TimescaleDB

- Web: https://www.timescale.com/
- Documentation: https://docs.timescale.com/

InfluxDB

- Web: https://www.influxdata.com/
- Documentation: https://docs.influxdata.com/

Suricata

- Web: https://suricata.io/
- Documentation: https://docs.suricata.io/

Snort

- Web: https://www.snort.org/
- Documentation: https://docs.snort.org/

Wazuh

- Web: https://wazuh.com/
- Documentation: https://documentation.wazuh.com/

OSSEC

- Web: https://www.ossec.net/
- Documentation: https://ossec.github.io/docs/

Zeek (Bro)

- Web: https://zeek.org/
- Documentation: https://docs.zeek.org/