PROJECT

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

V4 result of the analysis of suitable components

KPB2 – Design of the SOCaaS service

type – documentation

delivery month – M3

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1. Introduction to the SOC as a Service design 3
2. SOCaaS components and their close interconnection 3
2.1 Detection, monitoring and analytical processing 3
2.2 Incident management, response and record-keeping 4
2.3 Supporting components and asset inventory 4
2.4 Interconnection of components and data flow 4
3. Selected components for the INNOSOC SOCaaS 5
3.1 SIEM: Evaluation and results of the analyses 5
3.1.1 Role of the component 5
3.1.2 Solutions assessed 5
3.1.3 Detailed requirements analysis 5
3.1.4 Analysis of key requirements: 7
3.1.5 Selection conclusion 7
3.2 SOAR: Evaluation and results of the analyses 7
3.2.1 Role of the component 7
3.2.2 Solutions assessed 7
3.2.3 Solution profiles 7
3.2.4 Detailed requirements analysis 9
3.2.5 Analysis of key requirements 9
3.2.6 Selection conclusion 9
3.3 EDR/XDR: Evaluation and results of the analyses 9
3.4 Vulnerability scanner: Evaluation and results of the analyses 11
3.5 CTI: Evaluation and results of the analyses 14
3.5.1 Role of the component 14
3.5.2 Solutions assessed 14
3.5.3 Solution profiles 15
3.5.4 Detailed requirements analysis 15
3.5.5 Analysis of key requirements 17
3.5.6 Selection conclusion 18
3.6 Asset inventory: Evaluation and results of the analyses 18
3.6.1 Role of the component 18
3.6.2 Solutions assessed 18
3.6.3 Solution profiles 19
3.6.4 Detailed requirements analysis 19
3.6.5 Analysis of key requirements 20
3.6.6 Selection conclusion 20
3.7 Management of cybersecurity incidents 20
3.7.1 Ticketing system 21
3.7.2 Incident recording and management 23
3.7.3 Reporting 25
3.7.4 Knowledge portal for the SOC team 27
Conclusion and recommendations 30
List of sources 31
Annexes .....
..... Chyba! Záložka nie je definovaná.

1. INTRODUCTION TO THE SOC AS A SERVICE DESIGN

The design of the SOC as a service (SOCaaS) solution within the INNOSOC project focuses on creating an elastic, multi-tenant and modular service providing advanced monitoring, detection of and response to cyber incidents. In line with specific objective No. 3 of the project, the aim of this document is to identify and recommend the technological components that will form the core of the proposed SOCaaS solution.

The identification of the individual modules was based on:

- The process model of the SOC centre,
- The needs of real-world deployment for multiple customers with different levels of security,
- Open-source and free solutions (for reasons of scalability and availability),
- The experience gained from testing, laboratory deployments and previous research of the research team.

2. SOCAAS COMPONENTS AND THEIR CLOSE INTERCONNECTION

A modern SOCaaS is an interconnected ecosystem of security technologies that must jointly ensure not only incident detection and response, but also operational stability, process transparency and long-term auditability. In designing the architecture of the SOCaaS solution in the INNOSOC project, we focused on selecting components that are flexible, scalable, community-supported and, at the same time, sufficiently mature for operation in multi-tenant mode.

The components were divided into three main areas corresponding to the phases of the security cycle – monitoring, response and record-keeping.

2.1 Detection, monitoring and analytical processing

This layer provides the basic function of every SOC – the collection, processing and evaluation of security data. It contains the following components:

- **SIEM (Security Information and Event Management):** the central analytical unit of the system, responsible for log collection, event correlation, anomaly detection, alert generation and analytical dashboards.
- **EDR/XDR (Endpoint/Extended Detection and Response):** threat detection at the endpoint level, their behaviour, files, processes, IOC registration, telemetry.
- **CTI (Cyber Threat Intelligence):** data enrichment through threat information (indicators of compromise, TTPs, attacker profiles), collection from both external and internal sources, categorization and normalization of data.
- **Vulnerability scanner:** identification of weaknesses in systems and applications, detection of outdated software, misconfigurations or exposed services.

The components of this layer are interconnected – outputs from EDR are correlated in the SIEM, CTI provides context that improves detection quality, and vulnerabilities complement the risk assessment.

2.2 Incident management, response and record-keeping

Once a security event has been detected, it must be classified as an incident, escalated, documented and closed. This layer ensures practical operation and team coordination. It contains the following components:

- **Ticketing system:** a tool for recording all security incidents, changing their status, assigning them to analysts, escalating them according to severity level, SLAs and full auditability of handling.
- **Incident recording and management:** case management of incidents, management of observables (IP, hash, URL...), categorization, incident templates, tracking of handling metrics and linking to other systems (e.g. SIEM, CTI).
- **SOAR (Security Orchestration, Automation and Response):** execution of automated responses, orchestration of multiple components when handling an incident (e.g. blocking IOCs, generating tickets, escalating alerts).
- **Reporting:** creation of overviews and reports for management, daily/monthly incident summaries, SOC workload status, sharing of outputs with customers and support for ad hoc data analysis.
- **Knowledge portal for the SOC team:** an internal knowledge base, playbooks, manuals, FAQs, incident handling guides and process procedures, available to all SOC members according to their roles and permissions.

This layer is crucial for the effective functioning of the entire SOCaaS. Incidents are not only handled but also systematically recorded, analysed and reused to improve detection and response.

2.3 Supporting components and asset inventory

Without a clear overview of the protected environment, it is not possible to ensure high-quality protection. A supporting layer was therefore also included in the design:

- **Asset inventory (Asset Management):** recording of hardware, software, virtual and cloud assets. Categorization, ownership, security status, location (rack, VLAN, IP range), connections and relationships between components.
- **Integration with other systems:** the ability to match assets with detected incidents, direct connection with SIEM, EDR, ticketing/vulnerability scanner.

This layer helps analysts to orient themselves quickly during incident handling and helps ensure prioritization and correct decision-making based on context.

2.4 Interconnection of components and data flow

The individual components are interconnected by means of APIs and interfaces that enable efficient data exchange and process automation. Example of an integration flow:

- The SIEM records an anomaly → creates an alert,
- The alert is automatically linked with CTI and its context is evaluated,
- SOAR initiates a response (e.g. blocking a hash/IP, generating a ticket),
- A ticket is created in the incident management system,
- The case is handled by the team, recorded and reported,
- Relevant findings are stored in the knowledge portal.

Such interconnection ensures the **responsiveness, coordination, trustworthiness and efficiency** of the entire SOCaaS solution.

3. SELECTED COMPONENTS FOR THE INNOSOC SOCAAS

3.1 SIEM: Evaluation and results of the analyses

3.1.1 Role of the component

The SIEM is the heart of the SOCaaS solution. It provides the aggregation, processing, correlation and analysis of security events from various sources in real time. It enables analysts to identify anomalies, generate alerts and escalate incidents, while increasing visibility over the client's infrastructure.

3.1.2 Solutions assessed

Based on the requirements defined in the criteria, the following were evaluated:

- **ELK Stack (Elasticsearch, Logstash, Kibana)**
- **Wazuh**
- **OpenSearch + OpenSearch Dashboards**
- **Graylog (Open)**
- **SIEMonster (Community Edition)**
- **Prelude OSS**
- **AlienVault OSSIM**

3.1.3 Detailed requirements analysis

This section summarizes how the individual solutions meet the defined requirements for SIEM components within SOCaaS. The evaluation is based on structured requirement categories and provides a more in-depth qualitative description for each tool separately.

ELK Stack (Elasticsearch, Logstash, Kibana)

ELK Stack provides extensive support for log collection using tools such as Filebeat, Winlogbeat or Auditbeat. These components enable the collection of logs from various types of systems, including Windows, Linux, network devices and application services. At the data analysis level, Logstash enables data normalization and parsing, while Elasticsearch provides a powerful indexing and query engine.

Although the free version itself does not provide advanced ML tools, basic alerting mechanisms can be set up via Watcher (or the Alerting plugin in the OpenSearch variant). Security functionalities such as TLS between components and RBAC are available only in the paid editions, which represents a limitation in an open-source environment. Kibana, however, provides intuitive and extensible dashboards with numerous visualization options and filters.

Wazuh

Wazuh integrates components for SIEM, EDR and FIM (File Integrity Monitoring). Data collection is possible via agents, which support logs from operating systems, files, audit mechanisms and application servers. In addition, it can receive logs from Syslog, which enables centralized data collection also from passive devices without an agent.

Wazuh provides its own correlation engine with numerous rules based on MITRE ATT&CK. These rules are regularly updated by the community. The web UI supports RBAC, task management and connection with Kibana dashboards. Its main strength is the combination of EDR features with a clear SIEM, which makes it an advantageous solution for smaller teams.

OpenSearch + Dashboards

As a fork of the Elastic Stack, OpenSearch is a fully fledged alternative to Elasticsearch, providing all key functions for log collection, processing and visualization. Beats and Logstash are compatible, which ensures broad log ingestion options.

ML Commons is a powerful plugin that provides free ML functionalities such as anomaly detection. In terms of security, it provides TLS encryption and RBAC out of the box. OpenSearch Dashboards enables intuitive display of metrics and creation of visualizations similarly to Kibana, but with a more open development model.

Graylog

Graylog supports data collection from various sources, but works best with formats such as Syslog, GELF and JSON. For log processing it uses a pipeline architecture that makes it possible to define processing rules (e.g. data masking, normalization).

The free version of Graylog does not include ML analysis, but it offers good correlation and alerting options. The dashboards are responsive and easy to configure. For smaller teams, the lower complexity of administration is also an advantage.

SIEMonster (Community Edition)

SIEMonster combines the advantages of several tools (Elastic Stack, Wazuh, TheHive, Cortex). As a result, it offers rich options for log collection, correlation, data enrichment (from CTI tools) and incident response. The Community edition, however, has a limit on the number of agents (e.g. 100) and a somewhat more complicated installation.

The main advantage of SIEMonster is the pre-configured interconnection of components, which saves considerable time. On the other hand, its maintenance and updates are more demanding and require knowledge across several subsystems.

Prelude OSS

Prelude takes a specific approach to the SIEM solution – it is built on the IDMEF format, which is not commonly used in modern ecosystems. Its log collection capabilities are limited, and it lacks support for commonly used formats or collection agents.

Its correlation engine is powerful and modular, but it is not supported by ML or behavioural detections. Documentation is limited and community development is not very active.

AlienVault OSSIM

OSSIM provides all basic SIEM functionalities – log collection, correlation and normalization. It uses its own agent and supports protocols such as Syslog, NetFlow and SNMP. It is a fairly robust solution, but its interface and documentation are outdated.

It lags significantly behind in ML and CTI integration. Alerts and dashboards are simple, suitable rather for testing or academic purposes.

3.1.4 Analysis of key requirements:

- **Log formats**: ELK and OpenSearch support a wide range of formats (syslog, JSON, CSV). Graylog is oriented towards syslog formats. Wazuh has strong format support via its agent and the Elastic Stack extension.
- **Collection of audit and system logs**: Auditbeat (ELK, OpenSearch) provides direct mechanisms. Wazuh enables detailed auditing including file integrity (FIM). Graylog and OSSIM have limited capabilities without add-ons.
- **Analysis and detection**: The ML module in ELK is paid. OpenSearch has the ML Commons plugin. Wazuh provides basic behavioural detections. Event correlation is a strength of the Elastic Stack and Wazuh.
- **Scalability and performance**: ELK and OpenSearch are suitable for large environments, with support for horizontal scaling. Graylog is suitable for medium-sized deployments.
- **Integrations**: All solutions support integration with other tools via APIs, but ELK and OpenSearch have the best-documented support.

3.1.5 Selection conclusion

Recommended component: **ELK Stack** as the main SIEM tool, with the option of deployment via OpenSearch if a purely open-source licence is required. For small and medium-sized deployments, Wazuh is an excellent alternative, especially if it is used at the same time as the EDR component.

3.2 SOAR: Evaluation and results of the analyses

3.2.1 Role of the component

SOAR (Security Orchestration, Automation and Response) is the component responsible for automating incident responses, orchestrating security tools and supporting analysts' decision-making. In SOCaaS, it is a fundamental element enabling the scaling of response capacities without the need to increase staff numbers. It is indispensable especially when processing large volumes of alerts, where it reduces the MTTR and MTTC metrics.

3.2.2 Solutions assessed

- **TheHive + Cortex**
- **Shuffle SOAR**
- **StackStorm**
- **Tracecat (Fasten Health, ex-Kuiper)**
- **FIR (Fast Incident Response)**

3.2.3 Solution profiles

TheHive + Cortex

TheHive as an incident management system in combination with Cortex constitutes a powerful SOAR ecosystem. TheHive is used for incident management, categorization, notifications, evidence recording and auditing. Cortex is an integration module that enables the automatic execution of predefined analytical or response actions (so-called analysers). The

strengths of this pair are their modularity, API for integration and excellent documentation. Suitable for deployment in more complex SOCaaS environments.

Shuffle SOAR

Shuffle is a modern visual SOAR tool focused on simplicity and flexibility. It enables the creation of response workflows through a drag-and-drop interface, connection to dozens of services (Slack, Jira, Elastic, GitHub, VirusTotal) and orchestration of security actions without the need for coding. It supports asynchronous process execution, a debug mode and playbook versioning. However, it does not have integrated incident management and is more suitable as a complementary orchestrator to another system.

StackStorm

StackStorm provides a robust architecture for orchestrating events and response actions in IT and security systems. It works on the "event → trigger → rule → action" principle and supports hundreds of integrations (so-called packs). It enables the creation of complex playbooks, versioning and testing. Its generality is an advantage but at the same time a disadvantage – it is not focused exclusively on cybersecurity and requires more extensive configuration.

Tracecat (Fasten Health, ex-Kuiper)

Tracecat is an open-source SOAR tool focused on the intuitive assembly of security scenarios. It enables the creation of playbooks with connections to SIEM, EDR, e-mail notifications and event processing. Its advantages are simple deployment and its orientation towards security teams with a lower level of DevOps. The community is smaller, but the tool has potential for small and medium-sized organizations.

FIR (Fast Incident Response)

FIR focuses on incident management with an emphasis on classification, standardization and auditing. It makes it possible to define incident types, handling phases, severity levels, owners and response steps. It supports LDAP, REST API and data export. It is suitable as a basis for reporting and incident recording, and it can be connected with tools such as Cortex or StackStorm. TheHive is an open-source incident response platform that enables incident management, supports playbooks, creates relationships between events and also provides connection with other tools (e.g. MISP, Cortex). It has excellent documentation and community support. TheHive also supports multi-tenant environments, audit logs, automation via API and interactive IOC evaluation. Its strengths are modularity and the connection to Cortex for playbook execution.

Shuffle SOAR

Shuffle is a lightweight open-source SOAR tool with a modern web interface that enables the visual creation of workflows. Connection to the APIs of various tools (Elastic, Graylog, VirusTotal, Slack, Jira, etc.) is possible. Its advantages are ease of deployment (Docker) and flexibility in building custom actions. However, it does not support the same range of IR functionality as TheHive and does not have a natively prepared incident management model.

DFIR-IRIS

The IRIS (Incident Response Investigation System) platform is intended primarily for handling digital forensic analysis and incidents. It offers tools for the categorization, documentation and management of digital evidence. Although it is not directly a classic SOAR tool, it provides important functionality for automating the forensic handling of incidents and complements the functions of TheHive well.

Wazuh-integrated actions

Wazuh, as a security platform, provides some response options through active responses (e.g. IP blocking, scripts). However, it is not a fully fledged SOAR, so we regard this functionality as complementary, suitable especially in the absence of an external SOAR tool.

3.2.4 Detailed requirements analysis

TheHive + Cortex provides advanced incident management, categorization, auditing, evidence management and, via Cortex, also automated actions. It supports scalability, multi-tenancy and integration with a wide range of tools. It meets the requirements for playbook orchestration, API support and auditing.

Shuffle SOAR excels in the ease of creating and managing playbooks. It enables flexible integration using REST APIs and connectors. On the other hand, it lacks deeper incident management, but it is excellent as a complementary orchestrator for fast workflows.

StackStorm, as a universal automation tool, offers extensive flexibility, but requires knowledge of the concepts and structured programming of playbooks. Suitable for environments with a mature DevSecOps approach.

Tracecat covers the basic requirements for creating and operating response scenarios, but does not have a fully fledged incident management system. Its strength lies in its clarity, simplicity and the possibility of rapid deployment.

FIR is strong in incident documentation and management – it enables precise classification, life cycle phases, integration and auditing. However, it does not natively include response automation – it is suitable in combination with Cortex, StackStorm or other SOAR modules. a very good tool for recording, structuring and documenting incidents. It supports timelines and forensic attributes and enables the building of an incident knowledge base. However, it does not include a robust engine for automation or orchestration.

3.2.5 Analysis of key requirements

- **Playbook orchestration**: Best covered in Shuffle and Cortex (with TheHive). IRIS provides documentation, not active actions.
- **Incident lifecycle management**: TheHive and IRIS provide a full incident management model. Shuffle partially.
- **Integrations with other tools**: Shuffle and TheHive have strong APIs and numerous connectors. IRIS is more closed.
- **Auditing and reporting**: TheHive has a robust audit system. IRIS is strong in documentation. Shuffle has only basic auditing.

3.2.6 Selection conclusion

Recommended component: **TheHive + Cortex** as the main SOAR solution. They combine incident management, interactive evaluation and response automation. For simple scenarios or testing, Shuffle can be deployed as an alternative. We recommend IRIS as a complementary tool for the forensic handling of incidents.

3.3 EDR/XDR: Evaluation and results of the analyses

Role of the component:

EDR (Endpoint Detection and Response) and XDR (Extended Detection and Response) are the fundamental technological pillars for threat detection at the endpoint level. In the SOCaaS environment, they serve for the collection and correlation of, and response to,

events that would otherwise escape traditional perimeter detections. XDR extends the EDR concept with the ability to integrate data from networks, clouds, e-mail systems and other layers.

Solutions assessed:

- **Elastic Security**
- **Wazuh**
- **OSSEC**
- **OpenEDR**
- **Velociraptor**

Solution profiles:

Elastic Security

Elastic Security is an extension of the Elastic Stack that enables the detection and prevention of, and response to, security incidents. It uses Elastic Agent with integration into Fleet, provides extensive visibility over endpoints, threat detection capability via rules, support for MITRE ATT&CK and visualization through Kibana dashboards. Another strength is its ability to correlate data across multiple sources.

Wazuh

Wazuh, as a security platform, provides a combination of SIEM and EDR capabilities. In the EDR area, it enables file integrity monitoring (FIM), rootkit detection, activity auditing, collection of events from both the OS and applications, and support for MITRE ATT&CK. It supports both agent-based and agentless modes and provides a scalable architecture.

OSSEC

OSSEC is the predecessor of Wazuh and focuses on file change detection, log monitoring and basic incident response. It is lightweight and easy to deploy, but it lacks advanced correlation, modern integration interfaces and direct mapping to MITRE ATT&CK. Its advantages are low system requirements and a strong community base.

OpenEDR

OpenEDR is a solution developed by Comodo, focused on endpoint monitoring and analysis. It provides functions such as monitoring of processes, network connections, and changes in registries and file systems. It contains its own correlation engine and agent-based collection. Its disadvantages are limited documentation and a less active community.

Velociraptor

Velociraptor is a tool focused on threat hunting and forensic analysis. It enables detailed data collection, the use of queries in the VQL language, and real-time monitoring of memory and artefacts. It is not a classic EDR, but it is ideal for investigating incidents, retrospective analysis and detailed searching for compromises.

Detailed requirements analysis:

Elastic Security meets the requirements for data integration, correlation and visualization, has an advanced threat detection engine and an excellent interface for creating rules. It fully supports mapping to MITRE ATT&CK and enables direct response via Elastic Agent and its modular extensions.

Wazuh covers the requirements for FIM, log monitoring, rootkit detection and incident response. It is ideally usable in an environment requiring a transparent and easily auditable EDR system. Its strengths are scalability and community support.

OSSEC provides basic EDR capabilities suitable for simple deployments, but compared with Wazuh or Elastic Security it lags significantly behind in automation, orchestration and scaling. It does not include native connections to other SOCaaS modules.

OpenEDR meets some requirements for monitoring system activities, but its integration and flexibility lag behind. Its documentation and support are not on a par with competitors, which may be a risk when deploying in larger environments.

Velociraptor covers advanced requirements for forensics, threat hunting and retrospective analyses. It complements classic EDR solutions excellently, but does not fully fulfil the role of a preventive EDR.

Analysis of key requirements:

- **Detection of system state changes**: Elastic Security, Wazuh and OpenEDR provide advanced monitoring, OSSEC only basic monitoring.
- **Threat hunting and forensics**: Velociraptor dominates in this area. Elastic Security and Wazuh provide integrated analytical capabilities.
- **Incident response**: Elastic Security and Wazuh enable active response. OSSEC and OpenEDR only to a limited extent.
- **Scalability and integration**: Elastic and Wazuh offer rich integration options and horizontal scaling.

Selection conclusion:

Recommended components

- **Elastic Security** for organizations that already use the Elastic Stack and require a unified environment for collection, correlation and detection.
- **Wazuh** as a universal component for small and medium-sized deployments where stable community support and a transparent EDR are preferred.
- **Velociraptor** as a complement for specific case analyses and threat hunting. Suitable especially for SOC analysts when handling incidents.

Not recommended for the main deployment: OSSEC and OpenEDR, owing to their lower level of functionality, integration and support. **Recommended component**: **Wazuh as the primary EDR solution** for its agent-based architecture, extensive documentation, MITRE ATT&CK integration and ability to cover multiple levels of detection. **Supplementary component**: **Velociraptor** for advanced threat hunting and on-demand forensic analysis. **Falco** is recommended as a complement for cloud-native infrastructures and container deployments.

3.4 Vulnerability scanner: Evaluation and results of the analyses

Role of the component

The vulnerability scanner is a fundamental building block for ensuring a proactive approach in the SOCaaS architecture. It makes it possible to regularly identify weaknesses in the infrastructure, assess the status of updates, configurations and settings, and provide information for further prioritization and planning of corrective measures. The scanner outputs are often a source for SOAR systems, SLA metrics and reporting, as well as for assessing the client's security level.

Solutions assessed

- **Greenbone Community Edition (GCE)**
- **OpenSCAP**

Solution profiles:

Greenbone Community Edition (GCE)

GCE is the free open-source version of the solution from Greenbone Networks, based on the OpenVAS tool. It provides high-capacity scanning, extensive vulnerability databases via the community GVMD feed, the modern GSA web interface and support for scheduling, alerts and automation. It supports standards such as CVE and CPE, and enables the generation of reports in CSV, XML and PDF formats. Integration options include the GMP protocol, SSH, LDAP, RADIUS and others. GCE also enables a master-sensor architecture, which is advantageous for large distributed environments.

OpenSCAP

OpenSCAP is an open-source framework oriented towards auditing configurations and policies using the SCAP, OVAL, ARF and XCCDF standards. Its strength is compliance with standards (CIS, NIST, STIG), which predestines it for compliance audits. It supports agentless scanning and has low HW requirements, but it is primarily intended for individual hosts. It is suitable for auditing system settings and configurations, less so for bulk network scanning. The outputs are further processed via OpenSCAP Workbench or external visualization tools.

Detailed requirements analysis:

Licensing

Greenbone Community Edition (GCE) is a fully open tool that provides the functional core of a vulnerability detection system. Although it is a community version, it enables a smooth transition to the enterprise variant with additional features. **OpenSCAP** is completely open-source, maintained by the community as well as by organizations such as Red Hat, and provides a stable, transparent and auditable environment for the security analysis of configurations.

Scanning and coverage

GCE has significantly higher scalability in network scanning and enables in-depth verification of hundreds of IP addresses per day, depending on the performance of the host. It is suitable for enterprise-wide or multi-tenant scanning. Conversely, **OpenSCAP** specializes in individual systems and the assessment of their compliance status, and is therefore not efficient for large networks.

Vulnerability updates and support for standards

GCE uses the community feed (GVMD Data Feed), which is updated regularly. A commercial feed is available, which also allows for extension. Support for CVE, CPE and

CVSS is natively integrated. **OpenSCAP** uses the SCAP and OVAL standards; updates are platform-dependent and their management is often decentralized.

User interface and administration

GCE has GSA – an intuitive and modern web interface enabling scheduling, task tracking and report browsing. With **OpenSCAP**, most activities are carried out via the CLI or simple tools such as SCAP Workbench. This can be limiting for larger teams.

Agentless operation and scheduling

Both tools work without the need for agents, which simplifies their deployment. **GCE** has native support for scheduling, alerts, schedules and scan repetition, which enables efficient automation. **OpenSCAP** requires integration with external tools such as cron or Ansible.

Reporting and data export

GCE makes it possible to export results in multiple formats (CSV, PDF, XML, HTML), which simplifies their further processing. **OpenSCAP** generates outputs such as ARF, HTML and XML – these may be less usable without further conversion.

Deployment and architecture

GCE supports deployment as a container, a VM or classic on-premises. It supports a master/sensor architecture for distributed deployments. **OpenSCAP** is a monolithic tool and does not provide the option of centralized sensor management.

Performance and technical requirements

GCE requires at least 4 GB RAM, 2 vCPUs and an SSD. 8 GB RAM is recommended for higher performance. **OpenSCAP** is very lightweight and runs even on 1 vCPU and 1 GB RAM, which makes it an ideal solution for resource-limited devices.

Protocol support

GCE recognizes multiple network protocols, including DNS, DHCP, SNMP, SSH and others. **OpenSCAP** depends on defined SCAP policies, which may contain protocol parameters, but it is not primarily intended for network detection.

Integrations and scoring models

GCE supports the CVSS v3 standard, is preparing support for CVSS v4 and can be extended with EPSS using external feeds. **OpenSCAP** supports scoring via OVAL definitions. Integrations are more robust in GCE, with support for the GMP API.

Operational aspects and compliance with standards

GCE enables whitelisting and exception management. It supports standards such as ISO 27001, BSI and GDPR. **OpenSCAP** is very strong in the area of standards – it provides coverage of CIS, STIG and NIST and is suitable for regular audits and compliance assessment.

Analysis of key requirements:

- **Scanning scope**: GCE is suitable for large networks, OpenSCAP for individual systems.

- **Compliance with standards**: OpenSCAP is significantly stronger, suitable for compliance-centric environments.
- **Automation and scheduling**: GCE has built-in scheduling, OpenSCAP requires an external tool.
- **Suitability for SOCaaS**: GCE is more flexible and easier to integrate.

Selection conclusion

Recommended component: Greenbone Community Edition as the main vulnerability scanner for SOCaaS owing to its modularity, performance and open architecture. **Supplementary component**: OpenSCAP as a tool for compliance audits and system configurations according to standards (CIS, STIG, NIST).

3.5 CTI: Evaluation and results of the analyses

3.5.1 Role of the component

CTI (Cyber Threat Intelligence) is a key building block of the SOCaaS architecture, as it provides the ability to enrich detection rules, correlation mechanisms and playbooks with external threat information – including IOCs, TTPs, attacker profiling, campaigns and related vulnerabilities. In the multi-tenant SOCaaS service environment, it plays a role in:

- Distribution and management of threat feeds across tenants,
- Automatic enrichment of SIEM and EDR data with known threats,
- Creation of attack timelines and mapping to MITRE ATT&CK techniques,
- Supporting threat hunting and forensic analysis processes,
- Prioritization of incidents according to the threat context.

For the solution design, open-source platforms were chosen that cover various aspects of CTI – from IOC collection through visualization to STIX/TAXII compatibility.

3.5.2 Solutions assessed

On the basis of the preceding analytical phase and the categorized requirements, the following open-source solutions were selected:

- **OpenCTI** – a comprehensive platform for managing all levels of CTI (IOCs, TTPs, actors).
- **MISP** – a platform popular in the community for IOC sharing, strong in tactical CTI.
- **IntelMQ** – a framework focused on the automated collection, processing and distribution of IOCs.
- **YETI** – a database-oriented system for managing threat entities, aimed at analysts.
- **Vulmatch** – a tool focused on matching CVEs and IOCs, with an emphasis on CVSS and EPSS.
- **Threat Intel module (Elastic)** – an extension within the Elastic Stack for IOC visualization.

3.5.3 Solution profiles

OpenCTI is a robust tool supporting STIX 2.1 that enables the modelling of campaigns, TTPs, attacker groups and their mutual relationships. It enables integration with SIEM/EDR tools via the GraphQL API and provides graph-based visualization. It supports import from MISP, Abuse.ch and automated feeds.

MISP focuses mainly on IOC exchange and community data sharing. It enables semi-automated enrichment, a REST API, integration with corporate systems and export in various formats. It has an extension for MITRE ATT&CK and supports deduplication.

IntelMQ is an automation-oriented framework. Each "bot" has a specific task – collection, analysis, enrichment or output. It is ideal for automated feed processing and provides pipeline data processing.

YETI serves as a database aggregator of entities (IOCs, actors, campaigns). It enables both manual and API input, basic automation and simple visualization. It is suitable for analysts and threat hunters who need record-keeping and links between entities.

Vulmatch is a narrowly focused tool specializing in matching IOCs and CVEs, thereby enabling threats to be efficiently mapped to vulnerabilities. It provides simple outputs and prioritization according to CVSS/EPSS.

The **Threat Intel module in the Elastic Stack** enables basic storage of IOCs and their correlation with logs. It contains no CTI logic of its own, but through dashboards, visualizations and filter functions it can serve as a complement.

3.5.4 Detailed requirements analysis

This section assesses in detail six selected CTI solutions: **OpenCTI**, **MISP**, **IntelMQ**, **YETI**, **Vulmatch** and the **Threat Intel module (Elastic)** against all requirements defined in the categories of functional, technical and operational parameters. Each requirement is analysed separately for each tool, taking into account the requirements matrix provided in the analytical Excel workbook.

Licensing and general availability

- **OpenCTI** is fully open-source under the AGPL licence. It has a rich community and the option of enterprise support. Extension via connectors and the REST API is free of charge.
- **MISP** is open-source under GPLv3, with numerous add-ons and a large community. Thanks to its modularity, deployment is flexible.
- **IntelMQ** is a project under the auspices of CERT.at and ENISA. It is free of charge and its development is active.
- **YETI** is a community project, less active than OpenCTI, but available and extensible.
- **Vulmatch** is lightweight and focused on CVE–IOC matching. Open-source, but less well known.
- The **Elastic Threat Intel module** is available free of charge as part of the Elastic Stack; however, advanced features require the paid version (e.g. SIEM integration).

Types of CTI: strategic, operational, tactical

- **OpenCTI** excels in supporting all three layers of CTI – IOCs (tactical), TTPs and group profiling (operational), as well as strategic analysis through assessments, links and timelines.
- **MISP** handles the operational and tactical layers excellently – it manages IOCs and enables links between incidents and actors. The strategic layer is supported only marginally (manual notes).
- **IntelMQ** is a purely tactical CTI system – IOCs, URLs, IPs and their distribution. It does not support higher levels.
- **YETI** supports both IOCs and TTP profiling, but the strategic level is absent.
- **Vulmatch** does not work with TTPs or campaigns – it is intended only for CVE–IOC linking.
- The **Elastic module** enables IOCs to be recorded, but higher layers of CTI are not supported.

Data collection and integration with sources

- **OpenCTI** has the richest integration – it includes native connectors to Abuse.ch, OTX, MISP, VirusTotal and others. The connectors can be modified and scaled, and they are kept up to date.
- **MISP** has built-in modules for integration with sources such as Malware Bazaar, Abuse.ch and other community feeds. Integration is simple via cron/REST.
- **IntelMQ** has very broad data collection capabilities – each “bot” can be focused on a different source. Integration via bot-config or modules.
- **YETI** has several connectors, but they often require manual tuning. It is oriented more towards enrichment than collection.
- **Vulmatch** has limited support – it mostly depends on custom implementation.
- The **Elastic module** can index IOCs from sources via an ingest pipeline, but it has no CTI-centric connectors.

Data normalization and processing

- **OpenCTI** uses an entity-oriented model based on STIX 2.1. The data are readable, normalized and can be connected to other tools.
- **MISP** uses its own format, but supports STIX export/import. Normalization is strong and well documented.
- **IntelMQ** has a unified Message Format that can be converted – ideal for a pipeline.
- **YETI** normalizes through its own database of entities and tags. STIX support is missing.
- **Vulmatch** does not offer normalization – it is intended for a very specific purpose.
- The **Elastic module** enables IOCs to be standardized using its own fields, but this is not a CTI standard.

Automation, playbooks, deduplication

- **OpenCTI** provides entity deduplication, low-code automation using connectors and the possibility of linking with SOAR.
- **MISP** enables IOC deduplication by UUID and has rules for event-action processing.
- **IntelMQ** supports automated bots that can perform both deduplication and normalization – without a UI.
- **YETI** has limited support for automation, but enables IOC deduplication via hash.
- **Vulmatch** has no advanced deduplication.
- The **Elastic module** can enrich IOCs, but the deduplication logic is manual.

Integration, sandbox, visualization

- **OpenCTI** integrates via GraphQL and supports output feeds. It has built-in visualizations.
- **MISP** exports via API, REST or feed. Its visualizations are simple.
- **IntelMQ** has no visualizations – it integrates via JSON/syslog.
- **YETI** includes simple dashboards.
- **Vulmatch** does not include visualizations.
- The **Elastic module** uses Kibana – the visualizations are rich, but without TTP context.

3.5.5 Analysis of key requirements

On the basis of the categorized requirements, the following aspects were identified as the most important for the design of the SOCaaS service:

- **The ability to process all levels of CTI (strategic, operational, tactical)** – since the SOCaaS is to serve multiple customers, it is important to have a tool that provides a complete overview of threats – from IOCs to analytical reports.
- **STIX/TAXII compatibility** – interoperability between components and external partners is key for data exchange.
- **Deduplicated processing and normalization of data** – owing to the high frequency of feeds and the large volume of IOC data, it is essential to eliminate redundant records.
- **A high degree of automation and connectors to well-known CTI sources** – saves analysts time and increases the timeliness of the data.
- **Integration with SIEM and EDR** – automated enrichment of detection rules and alerts with CTI context is essential for the MTTD and MTTC metrics.

Compared against these requirements:

- **OpenCTI** clearly dominates – it covers all CTI levels, supports STIX/TAXII, offers a high degree of deduplication, modularity and excellent visualizations. It enables efficient connection to the other SOCaaS components.
- **MISP** is strong mainly in tactical CTI and has a large community and STIX support. It is very well suited as a complementary feed system.
- **IntelMQ** is a specialist in automated IOC collection, ideal as a pipeline for data pre-processing.
- **YETI** and the **Elastic module** are suitable as complements for analysts' work and data visualization, but are not sufficient on their own.
- **Vulmatch** has a narrowly specialized use and is not sufficient as the main CTI system.

3.5.6 Selection conclusion

On the basis of the evaluation of the functional, technical and operational requirements, **OpenCTI** was chosen as the main CTI component within the SOCaaS architecture. This tool offers the most comprehensive coverage of the requirements, enables flexible connection of other components (e.g. MISP, IntelMQ, EDR, SIEM), supports a high degree of automation and has rich visualization capabilities.

The following are recommended as complementary tools:

- **MISP** – for IOC management and community threat sharing,
- **IntelMQ** – as an automated pipeline framework for collecting IOC data.

Such a combination of tools will ensure full functionality for the entire threat management cycle, from collection to analytical interpretation.

3.6 Asset inventory: Evaluation and analysis results

3.6.1 Role of the component

Asset inventory is a fundamental pillar of every security solution. It enables the SOC team to gain an overview of all devices, system entities, applications and virtual or cloud resources that make up an organization's infrastructure. In a SOCaaS environment it plays an even more significant role, as it enables centralized management, categorization, monitoring and tracking of the security status of the infrastructure across different customers.

Vulnerability detection, incident contextualization, impact analysis and the configuration of effective ticketing are all based on good-quality asset records. Moreover, asset life cycle tracking supports recovery planning, audit processes and ensuring regulatory compliance.

3.6.2 Solutions assessed

On the basis of the defined requirements, we analysed the following open-source asset management tools:

- **LibreNMS** – primarily focused on network monitoring and SNMP discovery.
- **pi.Alert** – simple IP/MAC monitoring focused on home and small networks.
- **NetBox** – a modern IPAM and DCIM tool with advanced recording of networks and devices.
- **Ralph** – a full-fledged asset management system with CMDB and life cycle support.

- **GLPI** – a robust system with an integrated helpdesk and strong asset management support.

3.6.3 Solution profiles

LibreNMS

It focuses on SNMP-based monitoring with support for automatic device discovery. It offers basic tagging, device status display and data export. On the other hand, it lacks a full-fledged CMDB, life cycle management and clear relationships between assets. It is suitable as a complementary component for obtaining live data from the network.

pi.Alert

A minimalist solution suitable for small networks that provides a simple overview of connected devices based on IP and MAC addresses. Its capabilities are, however, limited – it provides no CMDB, categorization, links between assets or advanced management interfaces. The tool is suitable mainly for basic network visibility.

NetBox

Strong in the area of IPAM and DCIM. It enables links between assets, recording of both physical and logical topology, categorization and extensibility through custom attributes and Django plugins. Its advantages are a modern UI, a clear structure and the possibility of connecting to other systems via REST and GraphQL APIs. Its weakness is the lack of native asset discovery – it requires integration with external tools.

Ralph

Focused on asset life cycle management, with support for CMDB, categorization, change records and a REST API. It enables hardware, software and their status to be recorded throughout their entire operation. Its discovery functions are rather limited, but the record-keeping itself is very well designed. Ralph offers a good compromise between functionality and simplicity.

GLPI

The most comprehensive solution of the tools assessed. It includes excellent support for asset management, categorization, relationships between entities, planning, notifications and integration with other systems. With the help of the FusionInventory plugin, it supports discovery of assets in the network. A further advantage is the existence of a native ticketing module, which enables asset management and incident management to be combined in a single tool.

3.6.4 Detailed requirements analysis

LibreNMS offers mainly a network view – thanks to SNMP it can identify devices and monitor their status. However, it has gaps in asset management – it has neither a CMDB nor advanced categorization. Its advantages are integration with SIEM/monitoring systems such as Zabbix and powerful notifications.

pi.Alert is suited rather to basic IP visibility – it maintains an IP/MAC map and alerts on changes in the network. It cannot distinguish asset types, and it lacks both links between entities and a CMDB. It does not support the life cycle, and notifications are only basic.

NetBox provides precise network records – it can manage racks, VLANs, IP ranges, switches and server farms. It allows custom fields and relationships. However, it has no asset discovery, so data must be synchronized from other systems. Its advantages are its powerful API and clean UI.

Ralph supports CMDB, status management, life cycle tracking, audit and a REST API. It needs add-ons for discovery, but its asset management is comprehensive. Recording of OS, software and versions is possible. It also enables categorization, tags and owners.

GLPI is a robust tool – it has a CMDB, a ticketing system, notifications, life cycle support and detailed change logging. Numerous plugins extend the tool's capabilities. A further advantage is its connection with FusionInventory, which enables automated data collection. The UI is robust, yet easy to understand.

3.6.5 Analysis of key requirements

The analysis shows that the greatest emphasis should be placed on:

- The ability to manage the entire asset life cycle.
- CMDB support, categorization and links between assets.
- An API and the possibility of integration with other SOC components (e.g. SIEM, EDR).
- The ability to collect data automatically from the infrastructure.

GLPI and **NetBox** are the only ones that fully cover most of the key requirements. GLPI leads in asset management and ticketing, while NetBox dominates in network records and links between entities. Combined, they thus form a strong basis for asset management in a SOCaaS environment.

3.6.6 Selection conclusion

On the basis of the detailed comparison, we recommend deploying the following components:

- **GLPI** as the main asset management tool, covering CMDB management, record-keeping, life cycle, audit and integrations.
- **NetBox** as a complementary tool for precise management of the network infrastructure, IPAM and links between network components.

These tools complement each other and enable full visibility and control over the infrastructure, which is a basic prerequisite for effective detection and response within the SOCaaS service.

3.7 Management of cybersecurity incidents

Effective handling of cyber incidents is a key pillar of the SOCaaS architecture. It involves not only the mere recording of events, but a comprehensive cycle of their recording, management, response, reporting and knowledge retention. For the SOC team it is essential that incidents are managed systematically, with an emphasis on traceability, analytical work, coordination and knowledge support. An effective and coordinated response to security incidents is an indispensable pillar of every SOCaaS solution. Within the INNOSOC architecture, four subcomponents were identified that together form an integrated system for the management, recording, documentation and handling of incidents:

1. **Ticketing system**
2. **Incident recording and management**
3. **Reporting**
4. **Knowledge portal for the SOC team**

3.7.1 Ticketing system

Role of the subcomponent

The ticketing system serves as the central communication and organizational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

Solutions assessed

- **TheHive**
- **Zammad**
- **Request Tracker (RT)**
- **osTicket**
- **RTIR (RT for incident response)**

Solution profiles:

TheHive

TheHive is an open-source platform specifically designed for security incident management. Its advantage is deep integration with tools such as MISP (for Threat Intelligence) and Cortex (for data enrichment using analysers and responders). Thanks to this integration, TheHive can turn a received alert into a full-fledged incident (a so-called “case”) with numerous observables that can be sorted, filtered, enriched and assigned to individual analysts. Each case can contain multiple tasks, comments and attachments and is part of a workflow. The platform offers both a REST API and webhooks, making it ideal for integration with SIEM, EDR or automation systems. TheHive provides detailed audit logs and supports RBAC with granular permissions.

Zammad

Zammad is a modern ticketing system developed with an emphasis on usability and easy integration into an ITSM environment. It supports a fully responsive web interface with intuitive controls and drag-and-drop handling of attachments. Tickets can be created manually, via e-mail, via the API or using forms. Zammad enables flexible definition of queues, SLA rules, escalations and automated actions via so-called “triggers”. In addition to its native REST API, it also supports LDAP synchronization, SSO and connection to external systems such as SIEM or ticketing gateways. Tagging, templates, exports and multi-level department management functionality are built in. Although it is not a tool specifically focused on cyber incidents, its modularity and clean design make it a suitable option for supporting SOC processes in smaller teams or as a complementary helpdesk.

Request Tracker (RT)

RT is a robust and highly customizable system developed by Best Practical. Its strengths are stability, support for a wide range of extensions (including the RTIR incident response extension) and the ability to manage complex workflows in large organizations. RT is built on traditional ticketing logic, where requests are processed as queues that can be organized by teams, departments or customers. Within each ticket, comments, tasks, metadata, attachments and the history of changes are recorded. The system supports the creation of templates, role definition and SLA rules, and thanks to its REST API and CLI it can be integrated with other tools. RT is suitable for environments that require detailed request

management with the option of configuration according to internal processes; however, its user interface looks outdated and may be less intuitive for new users.

osTicket

osTicket is a simple yet reliable ticketing system with an open-source licence. It is suitable for smaller environments or as a support system for general communication with customers, including security requests. It supports the creation of tickets via the web, e-mail and API, and the configuration of auxiliary fields. It enables requests to be assigned to individual agents and SLAs and categories to be managed. osTicket supports exports, notifications, attachments and basic RBAC. The interface is clear, but less flexible than in more advanced solutions. Its advantage is low administrative overhead; its disadvantage is weaker integration with other security tools. In a SOC environment, osTicket is usable mainly as a basic platform for recording requests, but not as a full-fledged incident management system.

RTIR (RT for Incident Response)

RTIR is an extension of RT specifically designed for handling cyber incidents, which enables the use of predefined record types such as “Incident”, “Investigation”, “Block” or “Report”. These record types make it possible to track the course of an incident precisely, involve multiple teams, assign responsibilities and ensure an auditable trail. RTIR enables categorization, prioritization, templates, escalations and request processing across multiple teams. Its advantage is strong configurability and extensibility; its disadvantages are relatively complex administration and the need to customize the UI according to user preferences. Integrations with external CTI or SIEM systems are available via plugins or the API, but often require developer work. In the SOCaaS context, it represents a robust but less user-friendly solution.

Detailed requirements analysis

Basic functions and workflow: TheHive and Zammad offer an intuitive and responsive environment for ticket management. TheHive, however, is optimized for security cases and also includes attributes such as observables. RT/RTIR enables robust workflows, but the user interface is less modern. osTicket provides basic functions, but lacks deeper integration with security tools.

Notifications, prioritization, templates: Zammad and TheHive support automated notifications, different priority levels and ticket templates. RTIR offers SLA monitoring and categorization, but a less flexible UI.

Integrations and exports: TheHive dominates in the area of integrations (Cortex, MISP, SIEM). Zammad and RT enable API export and integration with external systems. osTicket has limited capabilities, and RTIR supports only some types of exports.

UX/UI and usability: Zammad offers a modern design, and TheHive is optimized for SOC teams. RT and osTicket have a more outdated look, but are functionally sufficient.

Analysis of key requirements

- **Multi-tenancy**: Supported mainly in TheHive and Zammad, less so in RTIR.
- **RBAC and auditing**: Strongly represented in TheHive, Zammad and RTIR.

- **Integration with SIEM/SOAR**: Strongly supported, particularly in TheHive.
- **Notifications and SLA**: TheHive, Zammad and RTIR all enable effective SLA monitoring and alerts.

Selection conclusion

TheHive is the clear winner, as it provides native support for incidents, is closely integrated with CTI and SOAR tools and has a modern interface. **Zammad** is suitable as an alternative for less complex environments or as a complementary helpdesk tool.

3.7.2 Incident recording and management

Role of the subcomponent

Incident recording and management is a core element of every security operations centre (SOC). This subcomponent ensures the systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A good-quality recording system must enable teams to collaborate in a coordinated manner, document steps, record related artefacts (observables) and link incidents to relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, audit, multi-tenancy and the possibility of retrospective investigation of incidents.

Solutions assessed

- **TheHive**
- **RTIR (Request Tracker for Incident Response)**

These solutions were selected for their focus on the security context, support for incident management processes, integration with other tools (CTI, SIEM, SOAR), open-source licensing and widespread adoption in the communities.

Solution profiles:

TheHive

TheHive is a leading open-source incident response platform that is widely used in practice for security case management. It enables the creation of so-called *cases*, which can contain tasks, comments, observables (e.g. IP addresses, hash values, URLs, domains) and many other artefacts. Observables can not only be recorded, but also automatically enriched using Cortex analysers or linked with MISP.

Each incident (case) can be further divided into tasks with a time frame, status, priority and assignment to an analyst. The system supports advanced categorization, custom case templates and workflows shared between organizational units. Among its main benefits are an intuitive web interface, a REST API, a multi-tenant architecture and strong customizability. The audit trail is captured at the level of actions, observables and comments. It enables the course of an incident investigation to be fully documented.

RTIR (Request Tracker for Incident Response)

RTIR is an extension of the stable Request Tracker (RT) system, specifically designed for security teams. It works with concepts such as *Incident*, *Investigation*, *Block* and *Report*, which makes it a well-structured tool for tracking and escalating incidents. Each incident can be divided into investigations, which are further recorded, commented on and tracked in detail.

The system enables cases to be linked using references, both manually and in a partially automated manner.

RTIR offers SLA management, RBAC, audit logs and export options. Although it is not modern in terms of UI, it is extremely stable, configurable and proven in large environments. Its case workflow is less flexible than in TheHive, but it integrates better with the formal processes of organizations that already use RT as their core helpdesk system.

Detailed requirements analysis

Case management:

- *TheHive* enables incidents to be created as cases, which can be fully broken down into tasks, comments, observables and metadata. A case can be associated with various entities – analysts, organizations, CTI records. Record-keeping is detailed and configurable.
- *RTIR* uses an approach based on typed requests (Incident, Investigation...), where each part has a specific structure and workflow. The level of detail is high, but case management is less visually focused and oriented more towards formal record-keeping.

Observables and their management:

- *TheHive* offers native management of observables with the ability to visualize, sort, share and enrich them.
- *RTIR* enables work with observables via add-ons or custom fields – however, this is not its native domain, which requires customization.

Categorization and templates:

- *TheHive* enables case templates according to incident type – e.g. phishing, malware, insider threat – which are pre-prepared with tasks and observables.
- *RTIR* uses formal request types – without automated templating for individual cases, but it allows the system to be extended with custom fields and workflows.

Audit, roles, multi-tenancy:

- *TheHive* has strong support for multi-tenancy, RBAC with various levels (observer, analyst, administrator) and a complete audit trail.
- *RTIR* offers access levels, departments, groups and very detailed auditing. Multi-tenancy is possible, but more difficult to configure.

Integrations and exports:

- *TheHive* provides a REST API, webhooks, and support for integration with SIEM, CTI (MISP), Cortex and reporting. Exports to CSV and PDF.
- *RTIR* supports integrations via REST API, CLI, scripts and plugins. Exports are possible to CSV and, in basic form, also to PDF.

Analysis of key requirements

- **Management of cases and observables:** TheHive clearly excels in the management of observables and complex incidents. RTIR keeps pace mainly in structured workflow cases.
- **Audit and activity logging:** Both tools provide detailed auditing. RTIR has the advantage of proven stability, TheHive of dynamism and visualization.

- **Integrations:** TheHive offers broader integration with security tools such as CTI platforms and SOAR components. RTIR is an open system, but requires more work for integrations.
- **Multi-tenancy:** TheHive offers more advanced features for isolating multiple teams/customers.

Selection conclusion

On the basis of its comprehensiveness, flexibility and modern approach, **TheHive** is the preferred solution for incident recording and management in the SOCaaS environment. It provides excellent integration and visualization capabilities and workflows oriented towards the security context. **RTIR** remains a suitable alternative for organizations that prefer a formal, stable infrastructure with proven processes, or that already use RT as their core system. For a multi-customer environment and scaling, however, a modern, API-oriented and visually refined tool such as TheHive is more advantageous.

3.7.3 Reporting

Role of the subcomponent

The reporting component within the SOCaaS architecture serves as the central means for the **aggregation, visualization and distribution of security information**. Its task is to ensure both regular and ad hoc outputs on the state of cybersecurity for various target groups – from SOC analysts, through team leaders, to the organization's management or clients. Such a reporting system must cover:

- Generation of metrics on incidents, alerts, SLAs and KPIs,
- Overviews of trends over time and evaluation of the performance of security measures,
- Adaptation of content according to the level of detail (technical vs. strategic),
- Automation and scheduling of the distribution of outputs,
- Retrospective analysis and archiving of historical data.

Effective reporting has a fundamental impact on **decision-making, accountability, audit** and **compliance** with regulations such as ISO 27001, GDPR or NIS2.

Solutions assessed

For this area, we assessed two proven open-source tools:

- **Kibana** – a native part of the Elastic Stack,
- **Grafana** – a general-purpose platform for visualization and monitoring from various sources.

Both solutions have strong communities, broad integration options and high flexibility in creating visualizations.

Solution profiles

Kibana is designed directly for the Elastic Stack and integrates naturally with Elasticsearch, Logstash, Beats, Wazuh and other components that are common in SOC architectures. It offers a rich set of visualization elements such as time series, heatmaps, metrics, charts and pivot tables. It is particularly suitable for analysing logs, SIEM alerts, events from Wazuh (XDR) or observables from Threat Intel modules.

Kibana's reporting module enables outputs to be generated in PDF and CSV formats, and with the paid X-Pack it is possible to schedule and automate the sending of reports (e.g. by e-mail or webhook). In open-source mode, this functionality can be simulated to a certain extent via scripts or the API. The interface is responsive and clear and supports interactive filtering. Kibana also supports RBAC and access auditing, which is important for multi-level teams.

Grafana is an exceptionally flexible platform for creating dashboards from many sources, including Elasticsearch, Loki, InfluxDB, PostgreSQL, SQL API and others. It enables the creation of very attractive visualizations suitable for management reporting. Thanks to a wide range of plugins (e.g. Image Renderer, Reporting plugin), it is possible to schedule and automate the export of outputs in PDF or PNG format. Grafana offers excellent customization of appearance, branding and embedding of panels, which makes it ideal for presenting outputs to clients or senior management.

Nevertheless, Grafana requires separate operation and configuration of integrations, and it is not fully natively linked with security functionalities such as SIEM, CTI, incident handling or XDR. This may lead to redundancy in integrations and increased operational demands.

Detailed requirements analysis

Both solutions provide broad data visualization and export capabilities. Kibana is stronger in technical and analytical visualizations of security data – it enables efficient work with Elastic indices, logs, SIEM alerts and Wazuh data. In addition, it enables automated detection using Elastic ML, linking with observables from CTI modules and detection rules from Wazuh. Report scheduling and distribution of results are available, although some features (e.g. scheduling of PDF exports) are tied to the commercial version.

Grafana excels in the visual quality of its dashboards and the ability to combine multiple sources in a single panel – for example, a combination of SIEM alerts, SLA metrics and infrastructure statistics. It is highly usable for SLA reporting and a simplified view of incidents from the perspective of the client or management. Its further advantages are easy deployment and broad compatibility with standard databases.

On the other hand, Grafana does not natively support Elastic's incident data model, is not linked with detection rules or observables, and its audit logs are less comprehensive. It is also less optimized for real-time full-text analysis of security data.

Analysis of key requirements

From the perspective of the INNOSOC SOCaaS architecture, the following properties are key:

- **Direct integration with SIEM and XDR:** Since we have selected **Elastic Stack** as the main SIEM and also plan to use **Wazuh as XDR**, we prefer a solution that fits into this environment without problems. Kibana is the natural choice in this case – it uses the same data backend, enables correlation of logs, incidents and IOCs, and provides a unified ecosystem.
- **Security and audit:** Kibana in combination with Elastic Security provides strong RBAC logic, access control, activity logging and the possibility of retrospective verification of every export or interaction. For a multi-customer SOCaaS environment, this is a critical requirement.
- **Extensibility:** Elastic and Kibana support further components such as Uptime, APM, ML and Watcher – which makes it possible to create incident-aware and SLA-aware

reporting based on intelligent detection. This reduces the need to duplicate data sources.

Selection conclusion

Although Grafana is a strong tool for generic visualization reporting and SLA overviews, in the context of the **Elastic Stack + Wazuh** architecture, **Kibana** is the logical and preferred choice for the reporting component.

Its strong native integration with security data, the ability to create incident-aware visualizations, and its support for observables, correlations and an audit trail, together with the Elastic SIEM and XDR (Wazuh) modules, form a consistent, scalable and technically clean solution.

For SOC analysts, Kibana provides deep insight into incidents and enables retrospective analysis and audit, while with suitable configuration it can also serve for management and client reporting. Therefore, in the design we prefer **Kibana as the primary reporting component for the INNOSOC SOCaaS platform**.

3.7.4 Knowledge portal for the SOC team

Role of the subcomponent

A knowledge portal for the SOC team is an important building block of every modern security operations centre. It serves as a **central repository of know-how, playbooks, procedures, recommendations and analytical guides**, and often also of historical lessons learned from incidents. A knowledge portal:

- Increases **the efficiency and consistency of the response** to incidents,
- Enables **rapid onboarding of new team members**,
- Supports **the standardization of processes and methodologies**,
- Enables the team to retain and share **proven procedures** and **its own analytical experience**,
- Strengthens the ability to **learn collectively from incidents** and build internal "threat intel" knowledge.

In a multi-customer environment, it is also important that the portal supports multi-level permissions, change history, document versioning and links to other SOC tools (ticketing, incident management, CTI).

Solutions assessed

In this area, we assessed open-source solutions that are commonly used as **wiki systems or knowledge base platforms**. The comparison included:

- **BookStack** – a modern and easy-to-use tool structured around books and pages,
- **Wiki.js** – a powerful, modular system with support for Markdown, Git, versioning and a wide range of integrations,
- **TheHive** – although not primarily a knowledge portal, it contains some components for case documentation and observables.

Solution profiles

BookStack is an open-source tool built on the Laravel framework that uses an intuitive **book–chapter–page** model. It supports document versioning, adding attachments, creating templates, hierarchical organization of content and RBAC at both book and user level. One advantage is an editor supporting both WYSIWYG and Markdown. It enables manual insertion of content and the creation of guides, IOC lists, playbooks and technical documentation. Integration is possible via a REST API, and export to PDF and Markdown is also supported.

Wiki.js is a powerful system using a modern UI, a database (e.g. PostgreSQL, MongoDB) and a Node.js backend. It excels in its support for **multiple editors (WYSIWYG, Markdown, Raw HTML)**, versioning, Git synchronization and support for document approval workflows. It is modularly extensible (e.g. PlantUML, Mermaid for drawing diagrams) and has detailed permissions and an audit trail. Integrations are possible via a REST API, webhooks and Git. In addition to documentation, it is also suitable as a **corporate wiki for the SOC team** with structured content and links to incidents or external tools.

TheHive has certain elements of a knowledge system – it enables observables, response templates and cases to be stored. However, its primary purpose is incident management. It does not offer a module for the separate management of structured documentation or playbooks. It is suited more to "incident-centric" notes than to a full-fledged knowledge base.

Detailed requirements analysis

In the area of **content management**, both BookStack and Wiki.js provide full support for document creation, editing, categorization, tagging and versioning. BookStack is simpler to use and excellent for team analysts who need to create playbooks or guides quickly. Wiki.js is more robust and suitable for integration with a development environment, with full synchronization to Git repositories, which also enables DevSecOps scenarios.

Search is a strength of Wiki.js, which uses ElasticSearch or PostgreSQL full-text search, with support for filters and categories. BookStack has a simpler but functional search engine. TheHive enables searching across observables and cases, but lacks systematic management of KB documents.

Linking with incidents is possible manually in both tools – e.g. by linking an incident ID or by category. In the case of TheHive, the linking is natural, but again – it lacks a broader documentation context (e.g. an updated guide for a new type of ransomware).

Export and security – both BookStack and Wiki.js enable export to PDF, Markdown and HTML. Wiki.js also enables a Git version of each article. RBAC is supported in both tools – definable in detail at the level of books, articles, teams or roles. Change auditing, version history and approval workflows are available mainly in Wiki.js.

Analysis of key requirements

The knowledge portal must be:

- **Intuitive and quick to use for SOC analysts** – BookStack is simpler in this respect, yet sufficiently powerful.
- **Sufficiently structured and technically mature** – Wiki.js enables comprehensive content management, revisions, integrations and security measures.
- **Extensible and integrated** – Wiki.js is better prepared for linking with ticketing, incident management, CI/CD and documentation workflows.

- **Secure and auditable** – both tools have RBAC and history, but Wiki.js provides a more detailed approach for a security-critical environment.

Selection conclusion

For the INNOSOC SOCaaS platform, we recommend implementing the knowledge portal by means of **Wiki.js**. This tool provides:

- Strong support for structured documentation,
- The possibility of integration with existing tools (ticketing, CTI, Git),
- A rich ecosystem for creating playbooks, handbooks, lessons learned and SOPs,
- Advanced permissions and workflows for content control.

BookStack may be a suitable alternative where a simpler and less technical solution is needed. **TheHive** continues to fulfil the role of an incident-centric database of observables and cases, but it does not replace the need for a separate knowledge system.

From the perspective of efficiency, scalability and integration with the other SOC components, **Wiki.js is the most suitable candidate for the knowledge portal for the SOC team**.

CONCLUSION AND RECOMMENDATIONS

The architecture of the proposed SOCaaS solution for the INNOSOC project rests on seven key technological components that cover the entire cycle of activities of a modern security operations centre: from data collection, detection and enrichment, through analytical processing, response and documentation, to retrospective analysis and support for knowledge management.

Summary of the preferred solutions:

- **SIEM:** On the basis of a comprehensive assessment, **Elastic Stack** was selected, which offers extensive functionality, scalability, strong community support and powerful integration with all the other components, including XDR, reporting and incident management.
- **SOAR:** The triple combination of **TheHive + Cortex + Shuffle** proved to be the best combination of functionality, extensibility and community support, enabling advanced incident management, response automation and the creation of playbooks with a high degree of flexibility.
- **EDR/XDR:** The resulting recommendation comprises **Wazuh** as the core EDR solution with native integration into the Elastic Stack and **Velociraptor** as a complement for advanced forensic data collection on endpoints.
- **Vulnerability scanner: Greenbone/OpenVAS** was recommended, which provides extensive vulnerability coverage, support for SCAP standards and an open architecture suitable for multi-tenant deployment.
- **CTI:** The strongest functionality for comprehensive processing of CTI data is provided by **OpenCTI**. In combination with **MISP** (for IOC management) and **IntelMQ** (automated collection and feeds), it forms a robust and scalable CTI module for SOCaaS.
- **Asset inventory: NetBox** proved to be the best choice in terms of functions, API support and integrations, complemented by **GLPI** for CMDB functionality and **Ralph** for asset life cycle management.
- **Incident handling:** The preferred solution was **TheHive** (for incident management and ticketing), **Kibana** (for reporting) and **Wiki.js** (as the knowledge portal). This set-up was also selected on the basis of its interconnection with the other modules and the possibility of integration via REST API.

Recommendations for implementation:

- **Modularity and integration:** All components were selected with an emphasis on an open architecture and an API-first approach, which allows them to be interconnected flexibly. It is important to build the SIEM as the central component (in this case Elastic Stack) and to ensure that all the other parts are connected to it.
- **Support for automation:** Components such as Cortex, IntelMQ, Wazuh and Shuffle provide low-code automation and process orchestration capabilities, which significantly reduce the workload on the analyst team.
- **Multi-tenancy and scalability:** All the recommended tools allow the separation of environments, customers or organizational units, which is essential for SOCaaS.
- **Emphasis on documentation and retrospective analysis:** The combination of Wiki.js and Kibana ensures the documentation of procedures as well as a visual link to incidents, which increases the efficiency of the teams and reduces the time needed to handle similar incidents in the future.

The document in this form represents the basic technical strategy for the design and construction of the SOCaaS service. The recommended components are proven, open-source, and at the same time cover the requirements for professional operation, security and interoperability.

LIST OF SOURCES

Primary tool documentation:
SIEM tools:
1. Elastic Stack Documentation – https://www.elastic.co/guide/
2. Wazuh Documentation – https://documentation.wazuh.com/
3. OpenSearch Project – https://opensearch.org/docs/latest/
4. Graylog Documentation – https://docs.graylog.org/
5. SIEMonster Community Edition – https://siemonster.com/
6. AlienVault OSSIM – https://cybersecurity.att.com/products/ossim
7. Prelude OSS Documentation – https://github.com/preludeorg/

SOAR tools:
8. TheHive Project – https://docs.thehive-project.org/
9. Cortex – https://docs.thehive-project.org/cortex/
10. Shuffle – https://shuffler.io/docs
11. StackStorm Documentation – https://docs.stackstorm.com/
12. FIR (Fast Incident Response) – https://github.com/certsocietegenerale/FIR
13. Tracecat (ex-Kuiper) – https://github.com/FastenHealth/Tracecat

EDR/XDR tools:
14. Velociraptor Documentation – https://docs.velociraptor.app/
15. OpenEDR by Comodo – https://github.com/ComodoSecurity/openedr
16. Wazuh (EDR extension) – see above
17. OSSEC Documentation – https://www.ossec.net/docs/
18. Elastic Security (XDR) – see the Elastic documentation above

Vulnerability scanning:
19. Greenbone GVM (OpenVAS) – https://docs.greenbone.net/
20. OpenSCAP – https://www.open-scap.org/documentation/
21. Nikto – https://cirt.net/Nikto2
22. Nmap – https://nmap.org/book/

CTI tools:
23. OpenCTI Documentation – https://www.opencti.io/en/docs/
24. MISP Project – https://www.misp-project.org/documentation/
25. IntelMQ – https://certtools.github.io/intelmq/
26. Yeti CTI Platform – https://yeti-platform.github.io/
27. Vulmatch – https://github.com/vulmatch/vulmatch
28. Elastic Threat Intel module – see the Elastic documentation

Asset inventory tools:
29. NetBox Documentation – https://netbox.dev/docs/
30. GLPI Project – https://glpi-project.org/
31. Ralph Documentation – https://ralph-ng.readthedocs.io/
32. LibreNMS – https://docs.librenms.org/
33. pi.Alert – https://github.com/pucherot/pi.alert

Incident handling / ticketing / reporting / knowledge base:
34. RTIR (Request Tracker for Incident Response) – https://bestpractical.com/rtir
35. Request Tracker (RT) – https://docs.bestpractical.com/rt/
36. osTicket – https://docs.osticket.com/
37. Zammad – https://docs.zammad.org/

38. Kibana (Elastic Reporting) – see the Elastic documentation
39. Grafana – https://grafana.com/docs/
40. BookStack – https://www.bookstackapp.com/docs/
41. Wiki.js – https://docs.requarks.io/

Supplementary scholarly sources:

1. Nguyen, K., et al. (2021). *A Review of Security Information and Event Management (SIEM) Systems*. Future Internet, 13(8), 204. https://doi.org/10.3390/fi13080204
2. Bhat, M. A., & Kumar, P. (2022). *Advanced SIEM for Enhanced Threat Detection Using Machine Learning*. IEEE CSR. https://doi.org/10.1109/CSR54599.2022.9850371
3. Munoz, M., Garcia, C., & Valdes, J. J. (2023). *ML-based Optimization for SOC Event Triage*. ACM AI-CyberSec Workshop. https://doi.org/10.1145/3576888.3602857
4. Al-Shaer, E., et al. (2021). *Autonomic Security Operations: Challenges and Research Opportunities*. IEEE TDSC. https://doi.org/10.1109/TDSC.2021.3127814
5. Akinrolabu, O. A., et al. (2020). *Security Operations Centre: A systematic study and open challenges*. Computers & Security, 92. https://doi.org/10.1016/j.cose.2020.101747
6. Zografopoulos, I., et al. (2023). *Threat Intelligence Platforms: Evaluation and Future Directions*. Journal of Cybersecurity, 9(1).
7. Lefèvre, M., et al. (2022). *Toward Open Threat Intelligence Integration: Evaluation of MISP and OpenCTI*. NATO CyCon.
8. Tannenbaum, B., et al. (2025). *Enhancing EDR Capabilities in SOC Architectures Using Open Source Stack*. Proceedings of the IEEE EuroS&P 2025.
9. Kovačević, N., et al. (2025). *Multitenant Architecture for Scalable SOC-as-a-Service Using Elastic and Wazuh*. Journal of Network and Computer Applications. (March 2025 issue, doi pending)
10. Li, F., et al. (2025). *Knowledge Engineering for Cybersecurity: Integrating KBs into SOC Workflows*. ACM Transactions on Privacy and Security, 28(2).