# SOCCRATES

SOC & CSIRT Response to Attacks & Threats based on attack defence graphs Evaluation Systems

# D2.3

# System Architecture & Interface Specification

# Final version

| | |
|---|---|
| Deliverable type: | Report |
| Contributing work packages: | WP2 SOCCRATES platform design |
| Due date of deliverable: | 31/08/2021 |
| Submission date: | 30/08/2021 |
| Dissemination level: | PU |

| | |
|---|---|
| Responsible organisation: | TNO |
| Editor: | Frank Fransen, Sebastiaan Tesink |
| Revision: | 1.0 |

| | |
|---|---|
| Abstract | This is the final version of the system architecture and interface specification. It describes all components of the full integrated version of the SOCCRATES platform, interfaces, and a description of how the platform is applied in each of the use cases. |
| Keywords: | Security Automation, System Architecture, SOC automation, CSIRT, Incident Response |

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 833481
Call H2020-SU-ICT-2018 • Innovation Action • Start date: September 1st, 2019

**Classification level: Public**

# Management summary

The main objective of SOCCRATES is to develop and implement a security automation and decision support platform for SOCs and CSIRTs that will significantly improve an organisation's capability to quickly and effectively detect and respond to new cyber threats and ongoing attacks. To validate the platform's capabilities, five use cases have been defined to be evaluated across two pilot sites.

This document describes the final specification of the system architecture and the components. This specification builds on the deliverable SOCCRATES D2.2 that described the initial system architecture and interface specification of the SOCCRATES platform. This document also updates how the SOCCRATES platform is applied within each of the five SOCCRATES use cases, that were initially explored in SOCCRATES deliverable D2.1.

Furthermore, this deliverable provides guidance on the deployment of the SOCCRATES platform. This includes deployment options, integration in existing environment, staffing and security. The deployment options include the support of an in-house SOC or Managed Security Service Provider (MSSP) based SOC, and on-premise deployment or (partly) cloud-based deployment. Since the SOCCRATES platform is able to oversee and control important elements of the IT infrastructure, the security of it is an important issue. A security analysis of the SOCCRATES Platform is presented and guidance is given for securing the deployment of the SOCCRATES Platform.

| | |
|---|---|
| Contributor(s) | Ewa Piatkowska (AIT)<br>Ruben Trapero (ATOS)<br>Jesus Villalobos (ATOS)<br>Ville Alkkiomäki (F-Secure)<br>Seamus Moloney (F-Secure)<br>Per Eliasson (Foreseeti)<br>Erik Ringdahl (Foreseeti)<br>Christophe Kiennert (IMT)<br>Marwan Lazrag (IMT)<br>Paul-Henri Mignot (IMT)<br>Mathias Ekstedt (KTH)<br>Preetam Mukherjee (KTH)<br>Ashish Dwivedi (KTH)<br>Martin Eian (mnemonic)<br>Piotr Kijewski (Shadowserver)<br>David Watson (Shadowserver)<br>Frank Fransen (TNO)<br>Sebastiaan Tesink (TNO) |

| | |
|---|---|
| Reviewer(s) | Marko Komssi (F-Secure) |

| **Security Assessment** | **See deliverables table for security assessment requirements** |
|---|---|
| Approval Date | 24/08/2021 |
| Remarks | Some minor changes required. Checked and approved for publication |

# TABLE OF CONTENTS


1 INTRODUCTION – RATIONALE OF THIS DOCUMENT ..... 6

1.1 The SOCCRATES project ..... 6
1.2 This deliverable ..... 7
1.3 Structure of this deliverable ..... 7

2 SOCCRATES PLATFORM OVERVIEW ..... 8

2.1 Introduction ..... 8
2.2 SOCCRATES component architecture ..... 10
2.3 Integration ..... 11

3 COMPONENT SPECIFICATION ..... 13

3.1 AI based Attack Detection ..... 13
*3.1.1 Functional specification ..... 13*
*3.1.2 API specification ..... 13*
*3.1.3 Data model ..... 16*
3.2 Orchestration & Integration Engine ..... 17
*3.2.1 Functional specification ..... 18*
*3.2.2 API specification ..... 18*
3.3 Infrastructure Modelling Component ..... 21
*3.3.1 Functional specification ..... 21*
*3.3.2 API specification ..... 23*
*3.3.3 Data model ..... 23*
3.4 Attack Defence Graph analyser ..... 25
*3.4.1 Functional specification ..... 25*
*3.4.2 API specification ..... 26*
*3.4.3 Data model ..... 28*
3.5 Course of Action Generator ..... 31
*3.5.1 Functional specification ..... 31*
*3.5.2 API specification ..... 31*
*3.5.3 Data model ..... 33*
3.6 Business Impact Analyser & Business Logic Modelling ..... 36
*3.6.1 Functional specification ..... 36*
*3.6.2 API specification ..... 36*
*3.6.3 Data model ..... 37*
3.7 Response Planner ..... 39
*3.7.1 Functional specification ..... 39*
*3.7.2 API specification ..... 40*
*3.7.3 Data model ..... 41*
3.8 Threat Intelligence Platform ..... 46
*3.8.1 Functional specification ..... 46*
*3.8.2 API specification ..... 46*
*3.8.3 Data model ..... 48*
3.9 Threat Data Collection & Threat Prediction ..... 49
*3.9.1 Functional specification ..... 50*
*3.9.2 API specification ..... 50*

3.10 Automated Reconfiguration ..... 58
*3.10.1 Functional specification ..... 58*
*3.10.2 API specification ..... 59*
3.11 Web Front End ..... 61
*3.11.1 Functional specification ..... 62*
4 USE CASES ..... 63
4.1 Introduction ..... 63
4.2 Use Case 1 ..... 63
*4.2.1 Detection ..... 64*
*4.2.2 Analysis ..... 65*
*4.2.3 Containment – Preliminary response ..... 67*
*4.2.4 Containment – Assess full extent ..... 70*
*4.2.5 Containment – Final containment ..... 72*
4.3 Use Case 2-5 ..... 75
5 SOCCRATES PLATFORM DEPLOYMENT ..... 78
5.1 Deployment options and considerations ..... 78
*5.1.1 Deployment options ..... 78*
*5.1.2 Deployment considerations ..... 79*
5.2 Security Analysis of the SOCCRATES Platform ..... 81
*5.2.1 Scope and assumptions ..... 81*
*5.2.2 Assets in SOCCRATES platform deployment ..... 82*
*5.2.3 Threats to SOCCRATES platform deployment ..... 83*
*5.2.4 Recommendations ..... 84*
6 REFERENCES ..... 86
7 ABBREVIATIONS ..... 87

# 1 Introduction – Rationale of this document

This section introduces the SOCCRATES project and defines the goals of this deliverable.

## 1.1 The SOCCRATES project

SOCCRATES (SOC & CSIRT Response to Attacks & Threats based on attack defence graphs Evaluation Systems) is an EU funded project under the Horizon2020 programme that has the following main challenge:

***How can SOC and CSIRT operations effectively improve their capability in detecting and managing response to complex cyber-attacks and emerging threats, in complex and continuously evolving ICT infrastructures while there is a shortage of qualified cybersecurity talent?***

The main objective of SOCCRATES is to develop and implement a security automation and decision support platform (‘**the SOCCRATES platform’)** that will significantly improve an organisation’s capability (usually implemented by a SOC and/or CSIRT) to quickly and effectively detect and respond to new cyber threats and ongoing attacks.

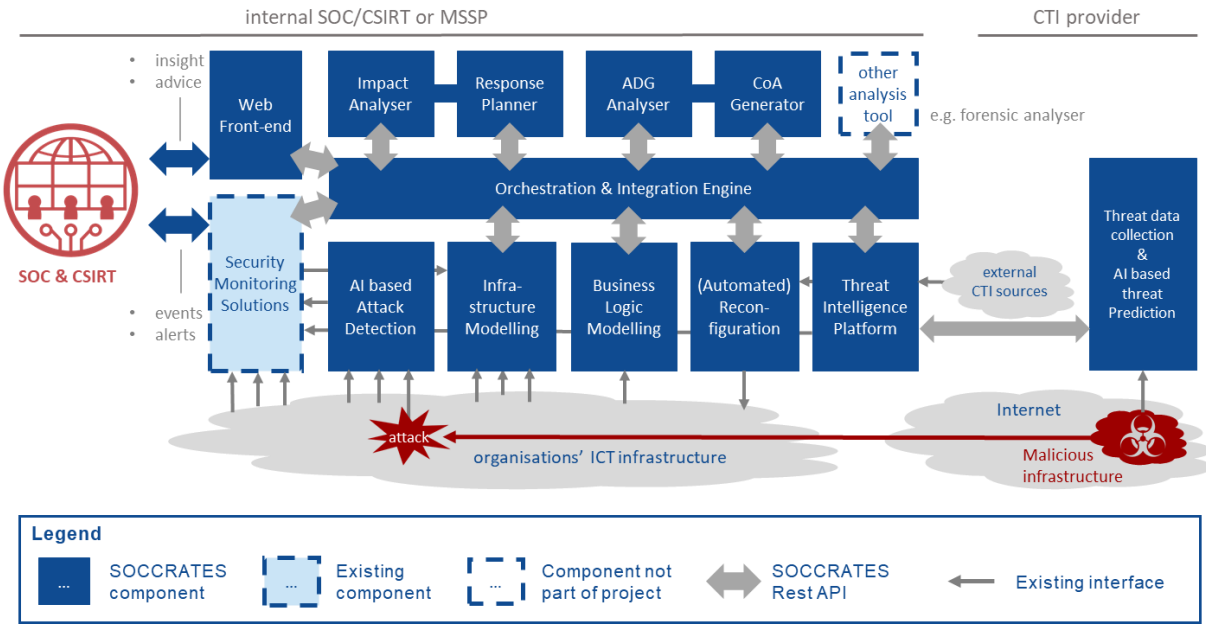


**Figure 1.1 – The SOCCRATES platform**

The SOCCRATES platform (see Figure 1.1) consists of an orchestrating function and a set of innovative components for automated infrastructure modelling, attack detection, cyber threat intelligence utilization, threat trend prediction, and automated analysis using attack defence graphs and business impact modelling to aid human analysis and decision making on response actions, and enable the execution of defensive actions at machine-speed.

SOCCRATES has the following concrete project objectives:

1. Deliver the SOCCRATES platform consisting of an orchestration function and a unique integration of innovative background solutions that seamlessly work together.
2. Show that the SOCCRATES platform can improve SOC operations by evaluating the SOCCRATES platform in two diverse real-life pilot environments.
3. Examine and illustrate the benefits of automation for selected SOC activities to help manage the cyber security skills gap in organizations.
4. Prepare for successful exploitation by the SOCCRATES partners of the individual innovated components and the integrated SOCCRATES platform in commercial products that are offered to the market and are available for the European (business) community.

Please visit www.soccrates.eu for more information on the SOCCRATES project.

## 1.2 This deliverable

In deliverable D2.1 [1] the SOCCRATES use cases have been analysed to identify the interactions between the different SOCCRATES platform components, and the data elements exchanged during these interactions. The initial specification of the SOCCRATES components have been described in deliverable D2.2 [2] including the plans for the development of each of these components. This deliverable contains the final specification of the SOCCRATES platform components and a description how they are used within each of the use cases. In addition, this deliverable also contains guidance for the deployment of SOCCRATES platform, including considerations for staffing and security of the SOCCRATES platform.

## 1.3 Structure of this deliverable

In chapter 2 of this deliverable an overview is given of the SOCCRATES platform and the use cases. In section 2.3 the integration of the components is described. Chapter 3 contains the specification of each of the SOCCRATES components. In chapter 4 the deployment of the SOCCRATES platform within the use cases is described. Finally, in chapter 5 deployment issues are discussed, including deployment options, integration into an existing environment, staffing and the security of the SOCCRATES platform is analysed.

# 2 SOCCRATES platform overview

## 2.1 Introduction

Five use cases have been defined to guide the development of the SOCCRATES platform. These use cases have been selected to represent the most relevant situations in which an organisation needs to reassess the security state of their ICT infrastructure, and determine if and how to react to protect the organisation's interests. The use cases are characterized by a particular security event that triggers the SOCCRATES platform to analyse and determine the best mitigation or response strategy.

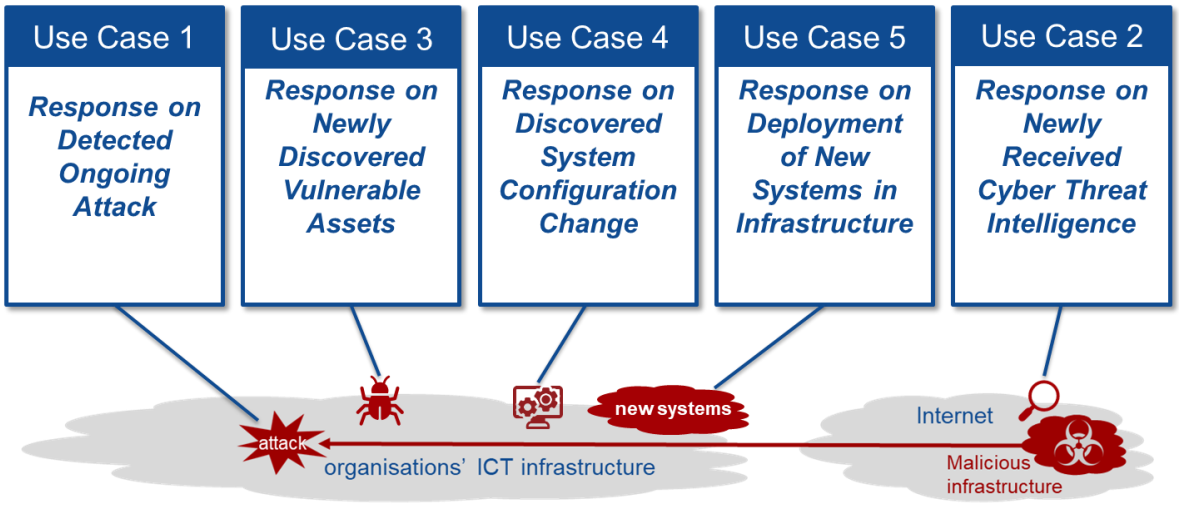


**Figure 2.1 – The SOCCRATES Use Cases**

- ***Use Case 1: Response on Detected Ongoing Attack***
  Detect ongoing attacks and automatically analyse the attack, automatically determine the best response, and initiate deployment of the selected response.
- ***Use Case 2: Response on Newly Received Cyber Threat Intelligence***
  Continuously collect new threat information, automatically analyse the potential business impact and determine best options for proactive mitigation.
- ***Use Case 3: Response on Newly Discovered Vulnerable Assets***
  Automatically detect vulnerabilities on assets in the ICT infrastructure, assess if they enable new attack paths, determine and initiate mitigation actions.
- ***Use Case 4: Response on Discovered System Configuration Change***
  Automatically detect configuration changes on assets in the ICT infrastructure, assess if they enable new attack paths and determine if action is needed.
- ***Use Case 5: Response on Deployment of New Systems in Infrastructure***
  Automatically detect introduction of new systems to the ICT infrastructure. Automatically assess the new situation and determine if security measures are needed.

There is a crucial difference between use case 1 and the other use cases. In use case 1 the organisation is responding to a detected ongoing attack. That means that an active adversary has access to the organisation's ICT infrastructure and can potentially cause lots of harm. An organisation must be very careful when responding to the attack, as this can tip off the attacker and triggering the attacker to cause more harm. For use case 1 we thus follow the incident response steps: detection, analysis, containment, eradication, and recovery, as described in NIST SP800 61 [3] and the ISO/IEC 27035

series [4]. As described in [1], SOCCRATES focusses on the automation of the first three steps of incident response: detection, analysis and containment.

In contrast to use case 1, the use cases 2 to 5 are triggered by security events that may point to a weakness or vulnerability in the security of the organisation's ICT infrastructure. As we assume that an adversary has not yet misused the (potential) weakness or vulnerability, the organisation can improve the security to mitigate the threat that an attacker will exploit the weakness or vulnerability. In other words, these use cases focus on preventing incidents and are focussed on increasing the cyber resilience of the organisation, see Figure 2.2.

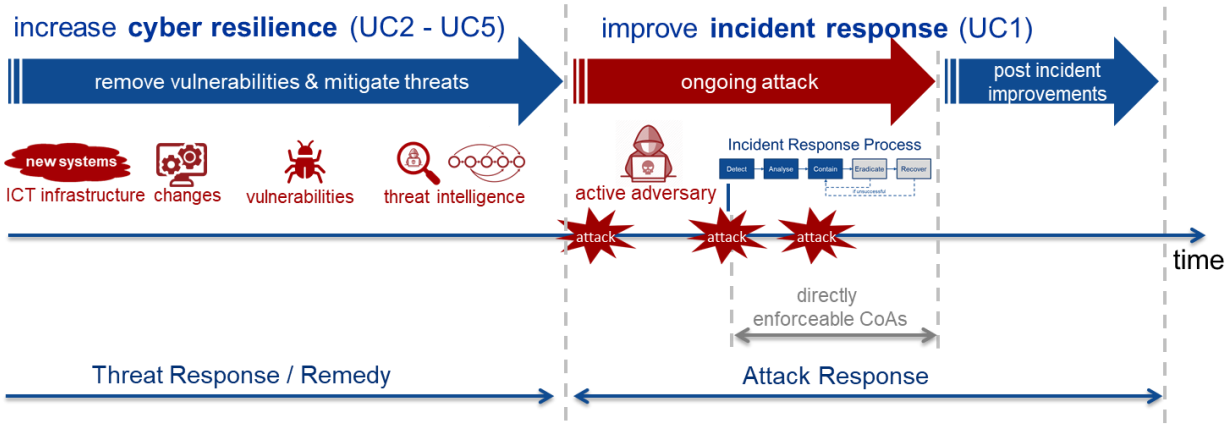


**Figure 2.2 – The SOCCRATES Use Cases along an incident response timeline.**

When analysing the automation of these use cases, four common phases can be distinguished that are inspired by the MAPE-K reference model used in autonomic computing [5] and self-adaptive systems. These phases are depicted in Figure 2.3 and can be summarised as follows:

1. *Monitoring phase* (M) – in this phase the system monitors security events specific to the use case, and triggers the orchestrating function of the SOCCRATES platform.
2. *Analysis phase* (A) – in this phase the SOCCRATES platform will automatically analyse the security event by collecting additional data, assessing the threat and determining the potential business impact. The results of the analyses is then presented as situational awareness to the SOC analyst. The SOC analyst may at this stage escalate the security incident to a CSIRT member.
3. *Mitigation & response Planning phase* (P) – in this phase the SOCCRATES platform will automatically generate courses of action (CoAs) and assess these based on an effectiveness and business trade-off (i.e. costs, operational impact). This is then presented as option awareness for the SOC analyst / CSIRT member.
4. *Mitigation & response Execution phase* (E) – in this phase the SOC analyst / CSIRT member has selected CoA and the SOCCRATES platform prepares and initiates the (semi-)automated execution of this CoA.

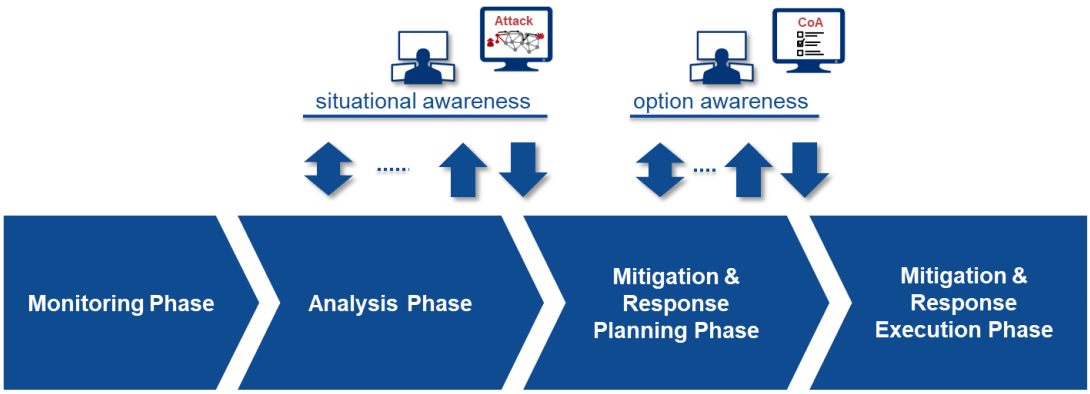


**Figure 2.3 – Illustration of four phases to automate the SOCCRATES Use Cases.**

The SOCCRATES platform uses an Orchestration and Integration Engine (OIE) to integrate, manage, and orchestrate all other components through these four phases. In the following section the SOCCRATES component architecture is further introduced.

## 2.2 SOCCRATES component architecture

Based on the analysis of the use cases the following capabilities for the SOCCRATES platform have been identified:

1. **Automated discovery and modelling** of an organisation's technical assets to provide an accurate, machine-readable description of the ICT infrastructure.
2. **Detection of (sophisticated) cyber-attacks**, with low false positives and improved understanding of the situation, by applying advanced data analytics techniques (including AI and deep learning) on large amounts of network and log data.
3. **Attack simulation by means of Attack Defence Graph (ADG)** based analysis on a model of the ICT infrastructure (both before and during an attack), and determination of the best response to these threats and attacks.
4. **Quantification of the (potential) business impact** of emerging threats and ongoing attacks by means of business impact modelling and appraisal of business trade-offs in possible mitigations.
5. **Cyber Threat Intelligence (CTI)** utilization in incident detection, analysis and response, including Adversary Emulation Plans to enable threat actor specific attack simulation with Attack Defence Graphs.
6. **Identification, classification, and prediction** of malicious activities and trends through automated analysis on large amounts of malicious infrastructure and malware forensic data.
7. **Plan and enable automatic execution of Courses of Action (CoAs)**, while including humans in the loop for authorization or manual reconfiguration, and supporting open standards for security automation.

These capabilities have been mapped to components in the SOCCRATES platform, depicted in Figure 1.1. The central component is the *Orchestration and Integration Engine* (OIE) that facilitates communication and exchange of information between components and orchestrates the actions undertaken by the SOCCRATES platform when it is triggered. The OIE consists of open source tools Activiti for Orchestration and Cortex for Integration. The OIE, including Activiti and Cortex, will be specified in section 3.2. The general concept of integration with Cortex within the SOCCRATES platform will be introduced first in section 2.3, as this concept applies to many of the SOCCRATES platform components.

Underneath the Orchestration and Integration Engine in Figure 1.1, all components are presented that directly interact with the ICT infrastructure of an organisation or collect (threat) intelligence from external sources (e.g. the Infrastructure Modelling Component and Threat Intelligence Platform). Above the Orchestration and Integration Engine the analysis components are shown (i.e. Impact Analyser & Response Planner, ADG Analyser & CoA Generator, and other analysis components). The SOC and CSIRT teams that are envisaged as primary users of the platform are shown to the left of the Orchestration and Integration Engine, as are the organisation's existing Security Monitoring Solutions.

The *Threat Data Collection & Threat Prediction* component (shown on the right) resides at an organisation that specialises in analysing malicious infrastructures. This component is thus outside the perimeter of the SOCCRATES platform itself, which is typically operated by a corporate SOC or MSSP. It does, however, interact closely with the SOCCRATES platform via the Threat Intelligence Platform (TIP), one of the native SOCCRATES platform components.

In chapter 3 each of the SOCCRATES components is specified.

## 2.3 Integration

As described in deliverable D6.1 [6], the OIE uses Cortex for integration with the other SOCCRATES components. Cortex is an open source project created by TheHive Project[1]. It was created to trigger analysis by different security tools without the need to interact with them individually. Cortex acts as a broker that communicates with those security tools, providing a common interface for any consumer entity that requests information from them. It allows to abstract consumer entities from the need to deal with complex APIs from different tools. Cortex provides a REST-API that can be used to interact with Cortex and invoke any of the security tools that are interconnected with it.

Cortex allows to interact with external tools through two types of connectors:

- *Analysers* allow to invoke external tools by requesting some analysis or action and providing some data in the request. In general, these analysers provide a response with a format that depends on the type of tool. Examples of analysers include requesting an antivirus scan for a file or the evaluation of risk for a given incident.
- *Responders* allow to trigger some action in an external tool, without the need of receiving any observable in the response. This connector is useful to trigger a certain response for an incident, such as sending emails to supervisors with relevant information, opening tickets in a response platform or trigger the direct enforcement of counter measures.

In Figure 2.4 a conceptual architecture is depicted of the OIE with the focus on the integration communication flow.

[1] https://github.com/TheHive-Project/Cortex

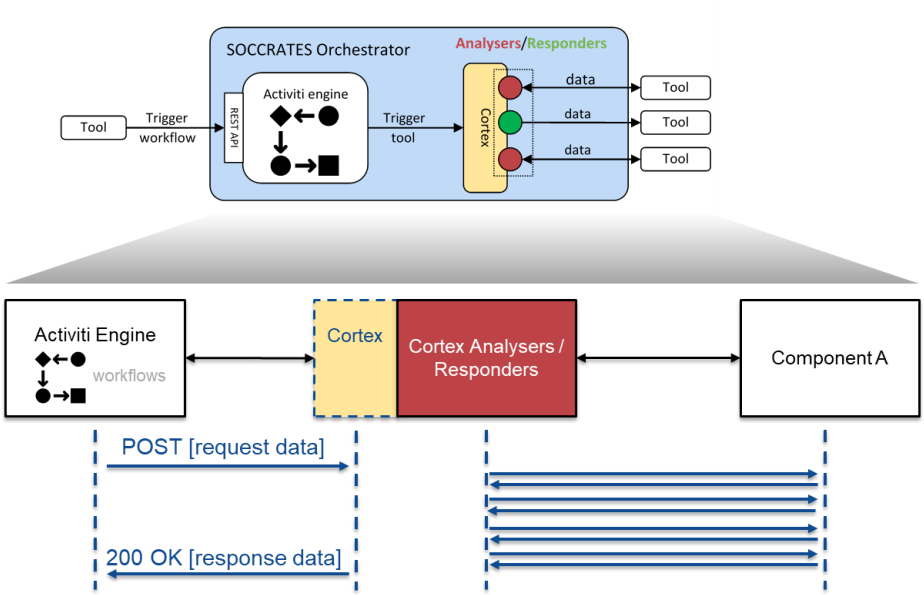


**Figure 2.4 – Conceptual architecture of the OIE with focus on integration.**

As part of the interaction with a security tool, it may be necessary for the Cortex analyser to interact with multiple tools, as depicted in Figure 2.5. This may for instance be to simplify the workflow in the OIE. Note that the access control from the OIE to the security tool is also handled by the Cortex analyser. In this deliverable we will specify the Cortex Analyser and/or Responder for each of the SOCCRATES components. If the component doesn't have a Cortex analyser, since the OIE does not directly interact with that component from the workflow, the API will be specified in this deliverable.

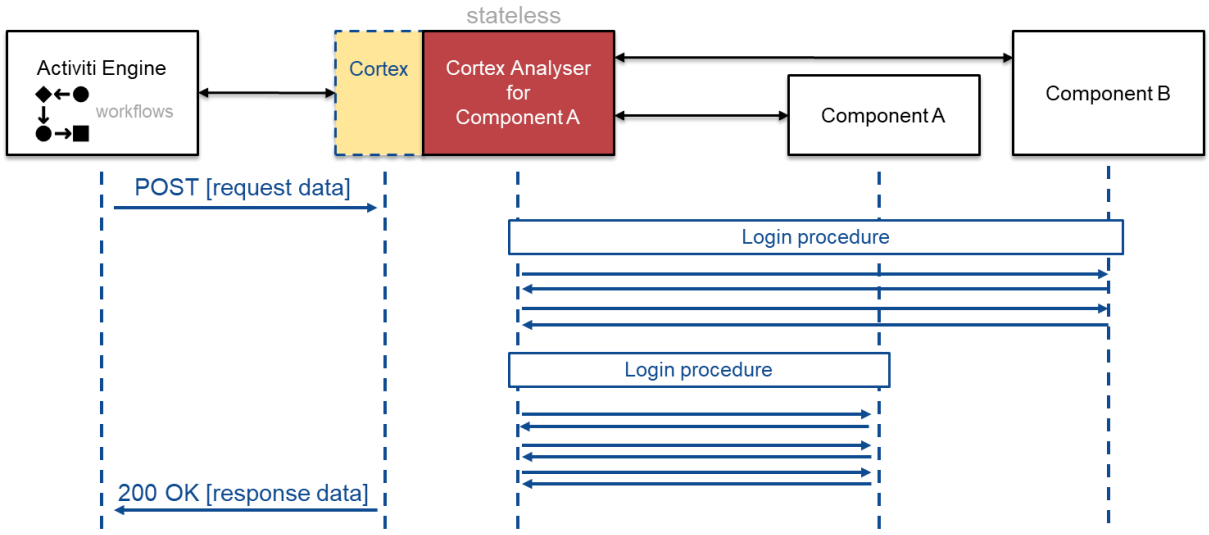


**Figure 2.5 – Communication flow of Cortex analyser that interacts with multiple tools.**

# 3 Component specification

In this chapter for each of the SOCCRATES components the functionality, API and data model will be specified. For some SOCCRATES components no specific data model has been specified, as the components use common existing data models.

## 3.1 AI based Attack Detection

### 3.1.1 Functional specification

The AI-based Attack Detection (AAD) component is intended to perform an attack detection based on multiple data sources. The AAD component leverages existing detection tools that provide different viewpoints on the state of the monitored system. In the current implementation, the following tools are used:

- *L-ADS* for netflow-based anomaly detection
- *ABC Tool* for internal host communication profiling
- *Aminer* for log data analysis
- *DNS Ninja* for analysis of dns traffic

The events reported by the above-mentioned detection tools are used for collective analysis in REASENS, to provide enriched interpretation of detected anomalies. In the current implementation, event-based reasoning is performed using evidential networks. More detailed overview of individual tools in the AAD component can be found in Deliverable D4.1 [7].

The objective of the AAD component is to determine the stages of an attack (or event types) to aid selection of an appropriate response action by the SOCCRATES platform. We focus on detection of three stages: initial access, lateral movement, and exfiltration.

Furthermore, integration of different detection capabilities is also intended to give more insight into attack characteristics and help reducing the number of false positives. The severity of the reported events (from REASENS) is adjusted to fit the collective meaning of detected anomalies and reliability of tools that reported them. That way, AAD may also improve the handling of weak indicators, i.e. true positives with low severity score, which might get overlooked and incorrectly dismissed.

### 3.1.2 API specification

#### 3.1.2.1 Integration

The deployment architecture of the AAD component is shown in **Figure 3.1**. Individual tools are deployed in Docker containers. Reported events are communicated via a message queue (Apache Kafka[2] is used in the current implementation).

[2] https://kafka.apache.org/

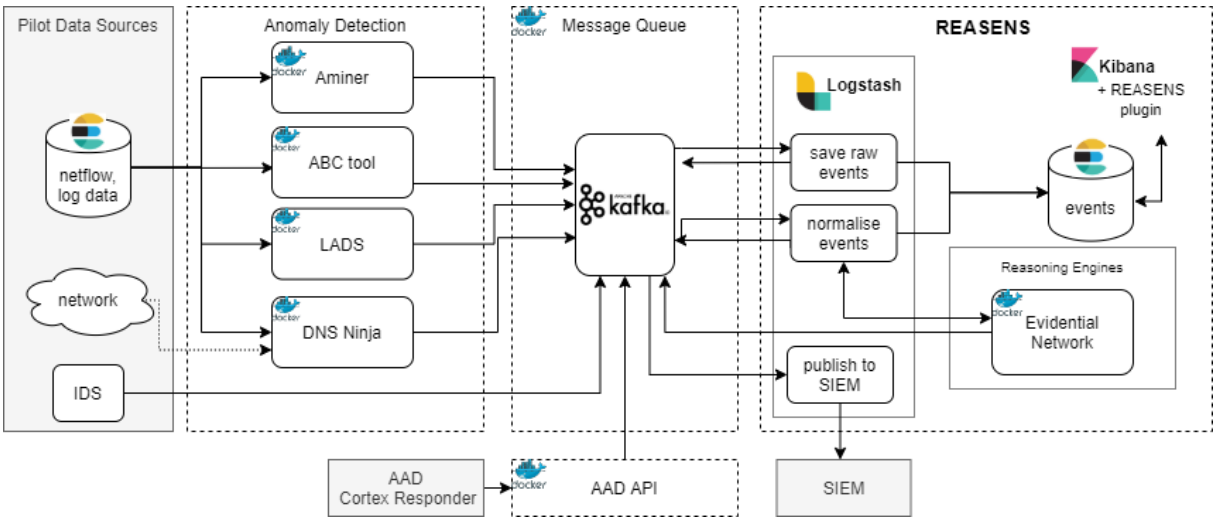


**Figure 3.1 – AAD deployment architecture.**

The AAD component has the following (external) interfaces:

- **Interface to pilot data sources**
  - In the current implementation this is achieved via Elasticsearch where relevant data sources are collected to be queried by multiple anomaly detection algorithms.
  - Some detection tools, e.g. DNS Ninja, might also tap directly into network traffic data which is another possible interface. However, for consistency, DNS Ninja also queries data from Elasticsearch in the current deployment.
  - In the future it will also be possible to forward the events from existing pilot detection tools (such as an IDS) directly to the Kafka message queue.
- **Interface to SIEM**
  - Events from detection tools and events from REASENS are pushed to Kafka and Logstash to be published (e.g. to the SIEM).
  - Data models for both output events are defined in section 3.1.3 (see Table 3.2, Table 3.3 and Table 3.4).
  - Considering consistency and ease of integration with different SIEM types, the AAD component will use syslog as output format.
  - In the future the Logstash syslog output plugin[3] will be used to implement this interface.
- **Interface to Cortex responder**
  - The purpose of this interface is to enable the OIE to inform AAD about changes in the infrastructure, such as new vulnerabilities (use case 3) or new assets (use case 5).
  - The interface, as shown in Figure 3.2 entails two components:
    - REST API to enable the OIE to trigger REASENS for relevant changes in the infrastructure.
    - Component to periodically query the REST API of the IMC to check the status of CVEs in the infrastructure.

### 3.1.2.2 API for Cortex responder

For integration of the AAD into the SOCCRATES Platform a REST API will be available to enable the OIE to trigger the AAD that a new vulnerability or new asset has been found. An AAD Cortex

[3] https://www.elastic.co/guide/en/logstash/current/plugins-outputs-syslog.html

responder has been developed to enable Activiti to interact with this REST API from AAD. The AAD will also have the capability to directly interact with the IMC via the REST API of the IMC. This is depicted in Figure 3.2.

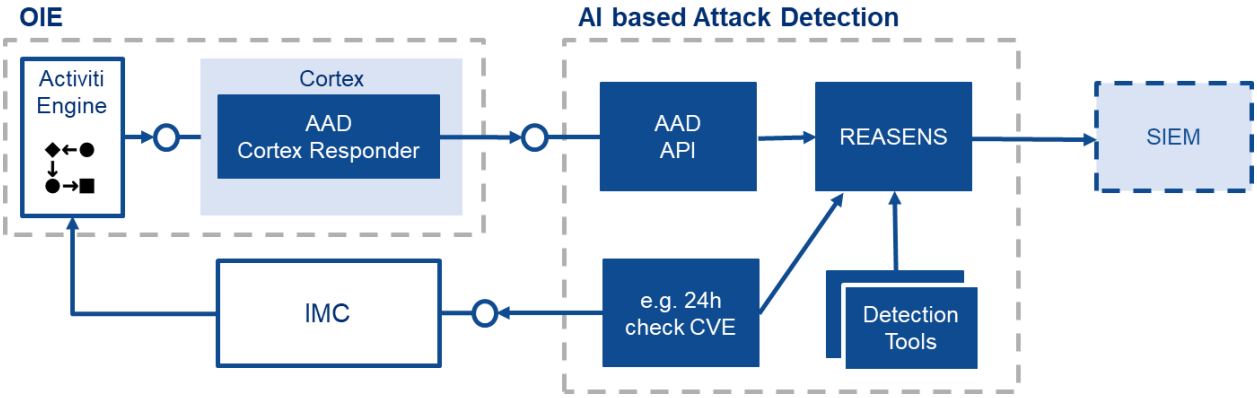


**Figure 3.2 – Integration architecture for AAD.**

The AAD Cortex responder will be triggered by the Activiti engine, given as input the following in-formation (message data format as defined in D6.1 [6]):

1. POST on new vulnerability found:

```
{
  "timestamp"    : "2021-05-08T18:04:12Z",
  "type" : "newVulnerability",
  "host" : "192.168.19.10",
  "vulnerabilities": [{
  "software": "Microsoft Edge (Chromium-based)",
  "cve": " CVE-2021-33741"
  }]
}
```

2. POST on new asset:

```
{
  "timestamp"    : "2021-05-08T18:04:12Z",
  "type" : "newAsset",
  "host" : "192.168.19.10",
}
```

Given the above input, the AAD Cortex responder will call a POST method AAD API with the parameters as specified in the table below.

**Table 3.1 – Specification of the AAD Cortex responder.**

| **AAD Cortex responder function** |
|---|
| **IMC_trigger**<br>Send new infrastructure data from IMC to AAD |
| **Sent parameters (POST)** |
| **timestamp** (required) string<br>time when infrastructure change has been detected, provided in known datetime format (e.g. UTC, ISO-8601, Unix Epoch) |

| |
|---|
| **type** (required) string<br>type of detected infrastructure change. Possible values: *newVulnerability*, *newAsset*<br><br>**host** (required) string<br>IP address of the host that the vulnerability has been found on<br><br>**vulnerabilities** (optional) array<br>An array of objects representing a vulnerability "cve" and "software" (that is affected by it). |
| **Response (Success)** |
| **ackMessage** string<br>Acknowledgment on success |
| **Response (Failure)** |
| **errorMessage** string<br>Error output message |

### 3.1.3 Data model

The AAD component provides two types of output to SIEM: (1) normalised events reported by detection tools and (2) enriched events from the REASENS engine. The data models for both of those types of events are defined in the tables below.

**Table 3.2 – Data model for events generated by the Evidential Network REASENS engine.**

| field | value (example) | description |
|---|---|---|
| **uID** | d81900c-7dab-4... | auto-generated unique identifier |
| **timestamp** | 1607610904.6240013 | date-time of event generation |
| **event source** | EN | detector (id, type, name or similar) |
| **event type** | lateral movement | initial access, lateral movement, exfiltration, ... |
| **mitreRef** | TA0008 | TA0001, TA0008, TA0010 |
| **event severity** | 0.87 | alert score, confidence of attack type, range from 0 to 1 |
| **system state** | {normal:0.34, attacked:0.66} | estimates for each system state |
| **Subevents** | [...] | uIDs of correlated events |
| **Metadata** | e.g. affected assets (hosts) | other information that could be provided, specific for the type of event |

Depending on the type of event, the metadata can have different forms. Currently, we consider three types of events (or event stages): initial access, lateral movement, and exfiltration. In the following, we list the examples of metadata for each of those types.

**Table 3.3 – Types of metadata available for different event types (tactics).**

| event type | metadata | description |
|---|---|---|
| Initial access | { src_IP, dest_IP} | Source and destination hosts |
| Lateral movement | [{src_IP, dest_IP}, …., {}] | List of affected assets (anomalous connections) |
| Exfiltration | {src_IP, dest_IP, flow_info} | Source and destination hosts with flow information e.g. bytes size, duration, etc. |

**Table 3.4 – Data model of normalised events generated by the detection tools.**

| field | value (example) | description |
|---|---|---|
| **uID** | d81900c-7dab-4... | auto-generated unique identifier |
| **timestamp** | 1607610904.6240013 | date-time of event generation |
| **event source** | abc_tool | detector (id, type, name or similar) |
| **event type** | Irregular_connection | Description of the anomaly |
| **event severity** | 0.87 | alert score, confidence of attack type, range from 0 to 1 |
| **host IP** | 192.168.220.15 | Source of the anomaly (host) |
| **Metadata** | e.g. flow details | other information that could be provided, specific for the type of event |

## 3.2 Orchestration & Integration Engine

The orchestrator and integration engine (OIE) is at the central core of the SOCCRATES architecture, brokering between the different tools available. Additionally, it provides the logic that guides the execution flow, which depends on the type of event processed. The OIE is easily extensible and adaptable to the characteristics of the infrastructure and the tools integrated in the SOC. It can also be triggered from different tools, depending on the event received. To this end, the OIE can perform different roles, depending on the event received which determines the execution flow to follow, namely reaction and prevention roles.

The reaction role happens when an event is received from a SIEM, denoting an incident that is currently ongoing in the infrastructure. The OIE orchestrates the tools to analyse the incident, contain it and determine the most appropriate course of action.

In a preventive scenario, the OIE can also play an important role when receiving events from tools that discover new detected vulnerabilities, emerging threats, new assets deployed in the infrastructure or in their configuration that might expose the infrastructure to potential new incidents. In such cases the OIE will invoke the required tools to analyse the impact of these events and determine potential actions that might be required to minimize the probability of some attack exploiting them.

### 3.2.1 Functional specification

As initially described in deliverable D6.1 [6] the OIE consists of two parts: the orchestration engine and the integration engine, see Figure 3.3. For the orchestration engine the open source BPMN workflow engine Activiti[4] is used, which executes the workflows created for SOCCRATES. It also allows to edit workflows using a Web GUI Editor. Activiti consists of three internal modules:

- The core of the engine, which executes the workflows,
- An admin module to manage installed workflows and know their execution state
- A REST API module which exposes the main operations related to the workflow, such as the management of the execution.

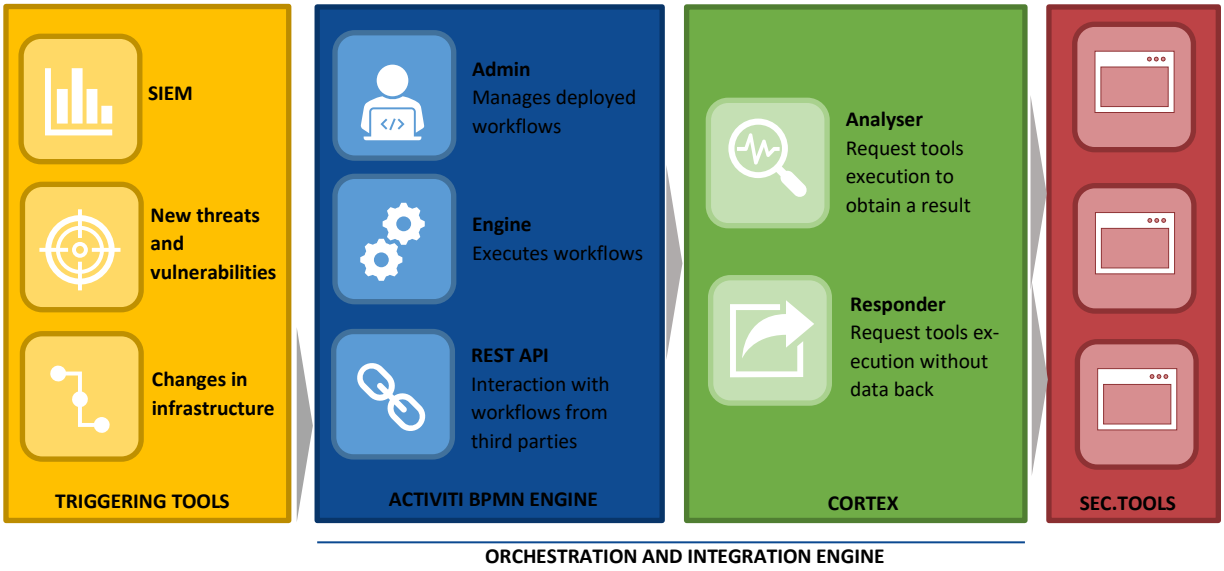


**Figure 3.3 – Elements involved in the Orchestration and Integration Engine**

The integration engine of the OIE is the module that acts as a common interface for the Activiti engine to interact with security tools. It is based on open source tool Cortex[5] from the TheHive Project[6] which abstracts the workflow engine from the varied complexity of interfaces exposed by security tools, including different technologies, data management or authentication issues. Through Cortex, the Activiti engine can invoke security tools in two ways:

- Using *Cortex analysers*, where a response is expected after the request. This is the case of tools invoking certain type of analysis returning the result in the analyser request.
- Using *Cortex responders*, where no response is expected after the request. This is the case of opening tickets in tracking platforms or sending emails, where no data returned is required.

### 3.2.2 API specification

Workflows in the OIE can be triggered through the Activiti REST API. In fact, the REST API allows to manage all aspects of the engine, including the management of the deployed workflows, the running processes, the available workflow models, manage the information related to executions, tasks or

[4] https://www.activiti.org/
[5] https://github.com/TheHive-Project/Cortex
[6] http://thehive-project.org/

management of user accounts among others. From all the options that are available in Activiti[7] the most important for SOCCRATES integration is triggering workflows (see Table 3.5).

**Table 3.5 – Specification of API endpoint that triggers a new workflow at the OIE**

| Endpoint | Triggers a new workflow | Method Type | POST |
|---|---|---|---|
| URL | activiti-rest/service/runtime/process-instances | | |
| Header options | Authorization:<br>String in the form "'Authorization: Basic <Base64AuthenticationToken>" that represents credentials to access to the API. The Authentication token is associated to allowed users.<br><br>Content Type:<br>String in the form: "'Content-Type: application/json", represents the format of the data included in the body of the request. | | |
| Body message | Message (string):<br>Name of the message that Activiti waits as input, which determines the workflow to trigger.<br><br>Variables (array):<br>Contains the information required by the workflow. It is composed of two additional fields:<br>- Name (String): Name of the input message<br>- Value (String): A string with the information required by the workflow, which depends on the type of message. It is a comma separated expression. | | |

According to Table 3.5, the body of the POST contains the message that the Activiti workflow expects. The type of message determines the workflow to trigger. It is also possible that a single POST request invokes several workflows in parallel in case all of them include the message name expected by each workflow. The following represents the POST request to trigger the workflow of Use Case 3, which receives a list of vulnerabilities found:

```
curl --location --request POST 'http://10.0.2.3:8080/activiti-rest/service/runtime/process-
instances' \
          --header 'Authorization: Basic <token>==' \
          --header 'Content-Type: application/json' \
          --data-raw '{
              "message": "input_vulnerability",
              "variables": [
                              {
                  "name": "input_vulnerability",
                  "value":"{\"Host\":\"computer01.soccrates.eu\",\"Soft-
ware\":\"cpe:2.3:a:google:chrome:*:*:*:*:*:*:*:*\",\"Vulnerability\":\"CVE-2021-21112\",
\"Date\":\"2020-12-12T16:45:14Z\"}"
                              }
                          ]
                      }'
```

Additional endpoints from the API will be used by the Web Front End to know the status of execution of workflows instances, obtain information such as the value of execution variables. The following is a list of potential endpoints that can be used for such purpose. Expected response examples are taken from the Activiti public specification.

[7] https://www.activiti.org/userguide/#_rest_api

Table 3.6 – API endpoints at the OIE to be used by the Web Front End

| Endpoint | Get a process instance | Type | GET |
|---|---|---|---|
| URL | runtime/process-instances/{processInstanceId} | | |
| Expected response | {<br>"id":"7",<br>"url":"http://localhost:8182/runtime/process-instances/7",<br>"businessKey":"myBusinessKey",<br>"suspended":false,<br>"processDefinitionUrl":"http://localhost:8182/repository/process-definitions/processOne%3A1%3A4",<br>"activityId":"processTask",<br>"tenantId": null<br>} | | |
| Endpoint | Get diagram for a process instance | Type | GET |
| URL | runtime/process-instances/{processInstanceId}/diagram | | |
| Expected response | {<br>"id":"7",<br>"url":"http://localhost:8182/runtime/process-instances/7",<br>"businessKey":"myBusinessKey",<br>"suspended":false,<br>"processDefinitionUrl":"http://localhost:8182/repository/process-definitions/processOne%3A1%3A4",<br>"activityId":"processTask"<br>} | | |
| Endpoint | List of variables for a process instance | Type | GET |
| URL | runtime/process-instances/{processInstanceId}/variables | | |
| Expected response | [<br>{<br>"name":"intProcVar",<br>"type":"integer",<br>"value":123,<br>"scope":"local"<br>},<br>{<br>"name":"byteArrayProcVar",<br>"type":"binary",<br>"value":null,<br>"valueUrl":"http://localhost:8182/runtime/process-instances/5/variables/byteArrayProcVar/data",<br>"scope":"local"<br>}<br>] | | |
| Endpoint | Get a variable for a process instance | Type | GET |
| URL | runtime/process-instances/{processInstanceId}/variables/{variableName} | | |
| Expected response | {<br>"name":"intProcVar",<br>"type":"integer",<br>"value":123,<br>"scope":"local"<br>} | | |
| Endpoint | List of process instances | Type | GET |
| URL | runtime/process-instances | | |
| Expected response | {<br>"data":[<br>{<br>"id":"7",<br>"url":"http://localhost:8182/runtime/process-instances/7", | | |

```
            "businessKey":"myBusinessKey",
            "suspended":false,
            "processDefinitionUrl":"http://localhost:8182/repository/pro-
cess-definitions/processOne%3A1%3A4",
            "activityId":"processTask",
            "tenantId" : null
        }

    ],
    "total":2,
    "start":0,
    "sort":"id",
    "order":"asc",
    "size":2
}
```


## 3.3 Infrastructure Modelling Component

### 3.3.1 Functional specification

The infrastructure modelling component (IMC) is a basis for producing a current and accurate model of the ICT infrastructure to enable automated security analysis tools to reason about the security state and for visualizations that accommodate human comprehension of the IT infrastructure. Figure 3.6 illustrates the IMC sub-components and their integrations with other components.

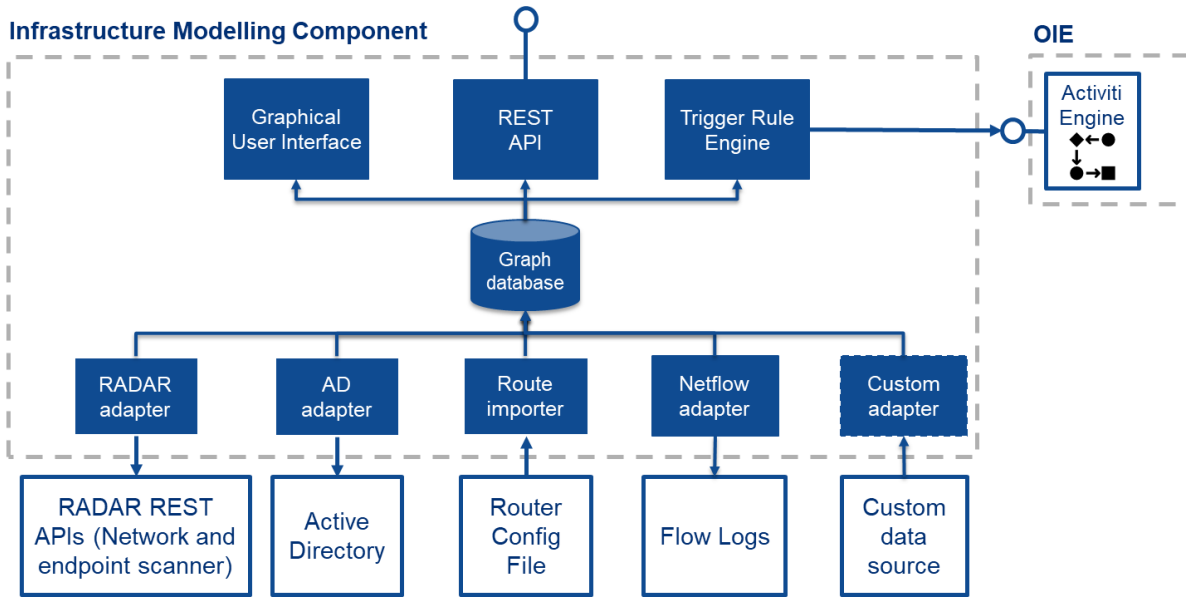


**Figure 3.4 – IMC integration architecture**

The graph-based central database in the Infrastructure Modelling Component will store the data sent by data source adapters and aggregate the partial infrastructure model updates coming from different data sources. The graph database stores data with timestamp, allowing data to be queried for a given period of time.

As a component, the IMC is self-contained and it does not depend on other SOCCRATES components to provide data. Instead it gets all data needed to maintain the infrastructure model from the data

sources through adapters. Separate adapters are needed for each data source to fetch and transform the raw data into the IMC model.

The IMC consists of the following sub-components:

- **Graph database**
  - Central storage of the IMC is based on the open source ACT platform from mnemonic.
  - Configured to use the IMC data model with schema files described in [8].
- **REST API**
  - A REST API allows other SOCCRATES components to query data from the IMC.
  - The technical implementation is based on mnemonic open source ACT platform.
- **Visualization**
  - The IMC has a graphical user interface allowing the infrastructure model to be visualized and queried by human users.
  - The technical implementation is based on mnemonic open source ACT platform.
- **Trigger rule engine**
  - The IMC has the capability to trigger the Orchestration and Integration Engine when predefined conditions are met in the graph database.
  - The IMC triggers the SOCCRATES use cases 3, 4 and 5, i.e. Response on Newly Discovered Vulnerable Assets, Response on Discovered System Configuration Change and Response on Deployment of New Systems in Infrastructure respectively.
  - The technical implementation is based on mnemonic open source ACT platform.
- **RADAR adapter**
  - The RADAR adapter is component built in Python, scheduled to query network and endpoint scanning results from RADAR and transform them into IMC data model.
  - The RADAR adapter uses standard RADAR REST APIs which have been extended during the project to support RADAR Endpoint Agent.
- **AD adapter**
  - The AD adapter is a Python component which is scheduled to query users, groups, organization units and their access rights for hosts in the network, extracted from Active Directory.
  - It uses an open source tool, called SharpHound, to fetch the data from Active Directory and convert the results into the IMC data model.
- **Route importer**
  - Import tool which can be used to translate and import static network routing and firewall rule settings into IMC.
- **Netflow adapter**
  - The Netflow adapter converts netflow data to the IMC. It ingests the total number of flows per day between two IP addresses to the IMC component. This provides other SOCCRATES components, such as the business analyser, a rough estimate of the business criticality and interdependency of assets for a given IP address. The adapter supports traffic flows from nfdump, ElasticSearch and AWS VPC Flow Logs.
  - For the AWS VPC Flow Logs, an AWS S3 bucket is used as intermediate storage of the logs.
- **Custom adapter for Vattenfall vulnerability scanning logs**
  - The installation of a secondary network vulnerability scanning solution (apart from F-Secure RADAR) at the pilot site of Vattenfall was not practical due the company security policies. Thus, a separate custom adapter is needed.

- Vattenfall will provide their existing vulnerability scanner output to populate the IMC via their own developed custom adapter.

### 3.3.2 API specification

The ACT platform provides a set of versioned REST APIs in order to interact with the IMC. The API endpoints are documented with Swagger, which contains a full description of each endpoint, including requests and responses. The Swagger documentation is available at https://act-eu1.mnemonic.no/swagger/.

Since the OIE does not directly interact with the IMC, no Cortex analysers have been developed for the IMC. The ADG Cortex Analyser, the BIA Cortex analyser and the Containment CoA Cortex analyser of the Response Planner use the APIs of the IMC to directly interact with the IMC.

The RADAR adapter uses a set of REST APIs provided by the RADAR Security Center, also known as F-Secure Elements Vulnerability Management, and documented with Swagger available through https://connect.f-secure.com/api-reference.

The AD Adapter uses SharpHound, an open source tool to query Active Directory. Documentation is available as part of the tool at https://github.com/BloodHoundAD/SharpHound3.

The Netflow adapter is able to read from nfdump files using nfdump (https://github.com/phaag/nfdump), from ElasticSearch (https://www.elastic.co/elasticsearch/) using its RESTful APIs and JSON (https://www.elastic.co/guide/en/elasticsearch/reference/current/docs.html), and AWS VPC Flow Log adapter uses the default log format documented at https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html#flow-logs-fields.

The custom Vattenfall adapter uses a proprietary JSON format and is not publicly available nor reusable outside Vattenfall environment.

### 3.3.3 Data model

The infrastructure data model is stored and modelled using the mnemonic ACT Object-Fact-Model. IMC data objects are modelled as objects and relationships between objects and modelled as facts in ACT platform. Furthermore, attributes related specific object are also modelled as facts, but linked only to the single object.

Figure 3.5 illustrates the IMC data objects and their relationships together with data sources used to populate the data.

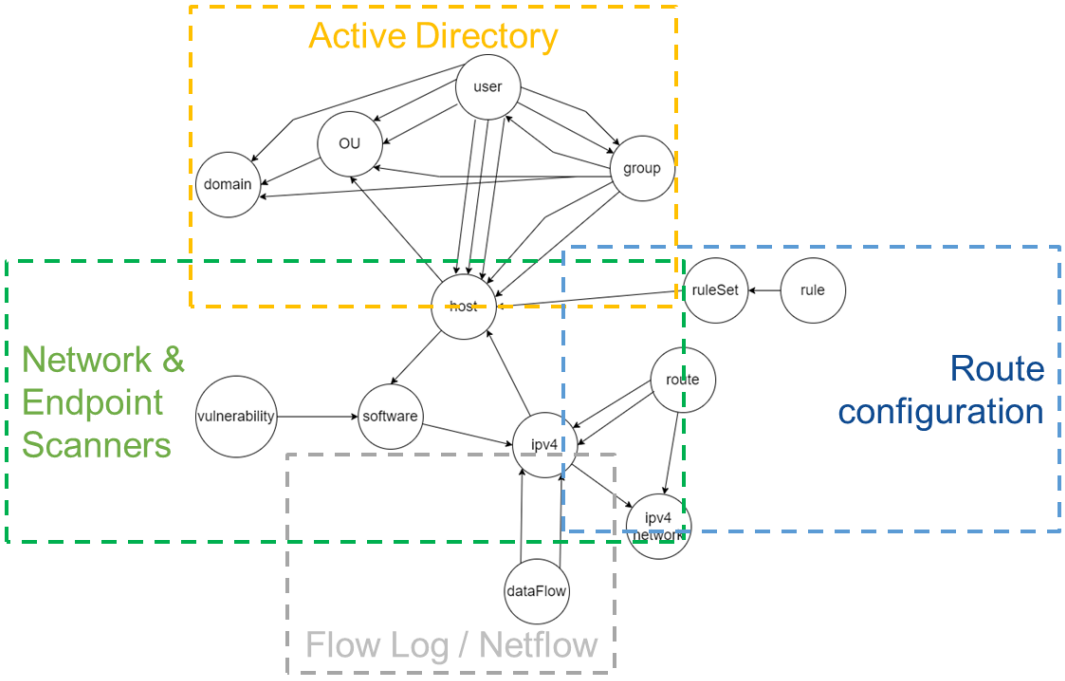


**Figure 3.5 – IMC Data model and data sources**

Each data source has a different view on the target ICT infrastructure and in addition to having different set of objects and attributes available, data sources see a different set of objects, which is illustrated with partially overlapping scoping of data sources. For example, Active Directory is only aware of the hosts, that have joined the domain, while a network scanner can see all devices in the internal company networks, whereas an endpoint scanner can further enrich host information with devices outside the company network.

The IMC data model and schema are described in detail in the D3.3 Specification of the common ICT infrastructure reference meta model [8]. Each adapter transforms and updates the graph database independently, linking it with data provided by other adapters using unique identifiers for shared objects.

Shared data objects are linked together into IMC data model as follows:

- **Host**
  - RADAR adapter stores and checks fully qualified name for each host and links the host with its current IPv4 addresses.
  - AD adapter stores and checks fully qualified name of the hosts and links access rights granted to the hosts for users, groups or organization units as facts.
- **IPv4 (address)**
  - IPv4 address, e.g. 10.255.129.123, is used as a unique identifier for IP addresses by RADAR, Flow Log and Route import adapters.
- **IPv4 network**
  - IPv4 network mask, e.g. 10.255.129.0/24, is used as a unique identifier for IP networks by RADAR, Flow Log and Route import adapters.

Duplicate IPv4 addresses or networks within single organization are not currently supported requiring the target network to follow best-practices in network design. With current implementation large

organization could be split into multiple sub-organizations with duplicate IP addresses, but deploying and managing multiple IMC adapters would be unpractical. A more practical approach would be to support namespacing for IPv4 addresses in all adapters and have namespaced configurations. Alternatively, IPv6 addresses could be used to differentiate duplicate IPv4 addresses. However, currently none of the pilot sites require duplicate IP addresses to be supported, so namespacing support has not been implemented yet.

## 3.4 Attack Defence Graph analyser

### 3.4.1 Functional specification

The ADG performs an analysis based on probabilistic attack simulations conducted on models of the infrastructure under assessment. The model is generated from data provided by the infrastructure modelling component (IMC) and can be further enriched with cyber threat information from the threat intelligence platform (TIP). Based on the analysis, ADG provides quantitative metrics on risk exposure and time to compromise (TTC), the most likely attack paths to high value assets as well as suggested mitigations.

The attack defence graph analyser will use the attack simulation and automated threat modelling solution called securiCAD. securiCAD is a foreseeti product, developed to perform virtual attack simulations on models of IT architectures, using automated threat modelling and artificial intelligence. Based on decades of research at the Royal Institute of Technology in Stockholm (KTH), securiCAD enables companies to transform their risk assessments from isolated technical problems to a holistic approach for quantifying risk exposure. securiCAD performs automated attack simulations on models of current and future IT architectures, identifies and quantifies risks holistically including structural vulnerabilities, and provides decision support based on the findings.

The attack simulations quantify a potential attacker's actions with probability of, and time to, success and most likely attack paths. securiCAD uses probability theory to represent and process uncertainty and assesses the time it takes for a highly capable adversary to compromise high value assets by generating attack defence graphs from a model of an IT architecture. The graphs are populated with probability distributions that specifies the amount of time it would take an attacker to traverse the steps in the attack graph. Monte Carlo simulations are then used to sample the distributions of the attack steps in the model. securiCAD provides insight to the attack times as a probability distribution – a Time to Compromise (TTC) - and most probable attack paths. Since all simulations are conducted on a model of the IT architecture, there is no impact on availability of the system under assessment.

A complete functional specification and system architecture is detailed in deliverable D3.1 Initial ADG based Attack prototypes [9].

The high-level functionality provided by the ADG are:

- Perform probabilistic attack simulation on a given infrastructure model provided by the IMC; i.e.
    - perform attack simulations for a particular set of high value assets and specific starting point and capabilities of the simulated attacker;
    - assess the probability of a successful attack with regards to specific security measures and model configurations;
    - find the most probable attack paths to the high value assets in the model.

- Quantify and provide a metric based on the most probable attack path to each of the high vale assets
    - A Time to compromise i.e., the time to a successful attack, will be provided for each high value asset.
- Compare and quantify the difference between simulation results with regards to the afore-mentioned metric. Difference with previous analysis could be due to:
    - New threat information from the TIP (use case 2);
    - Newly discovered vulnerable assets from the IMC (use case 3);
    - Change in the infrastructure model from the IMC (use case 4 and 5).
- Update or add new attack steps to the attack simulations based on for example increased use of a particular attack technique (use case 2.1.1).
- Update the parameters of vulnerabilities in the model based on new information about the vulnerability. For example, the availability of public exploits or released patches (Use Case 2.1.2).
- Define a threat actor with specific set of preferred attack techniques or capabilities (based on Adversary Emulation Plan) (use case 2.1.3).

### 3.4.2 API specification

securiCAD has a REST API that can be used for automation and programmatic interaction with the tool. However, this API is not designed for SOC and CSIRT integrations and features a lot of functionality that is not relevant nor necessary for the SOCCRATES platform. Furthermore, the REST API is not publicly available. As such, to create an open-source and general point of integration for the SOCCRATES platform it was decided that all communication from the OIE should go through a Cortex analyser. The ADG Cortex analyser will use the open-source Python SDK developed for securiCAD to provide integration with securiCAD as well as communication and data collection from other components. The ADG Cortex analyser will also be open-sourced and publicly available for anyone to use. Details of the securiCAD SDK can be found on GitHub[8].

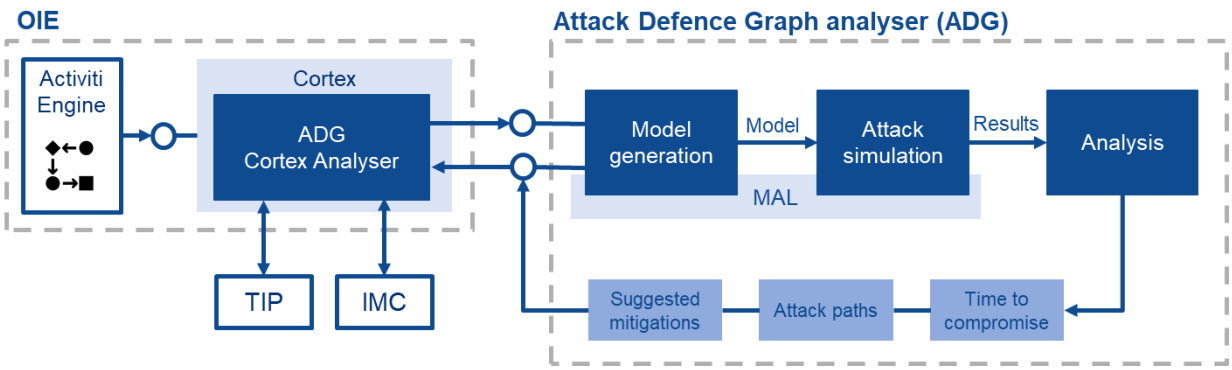


**Figure 3.6 – Integration architecture for ADG**

The ADG depends on the IMC for all Use Cases and the TIP for Use Case 2.

#### 3.4.2.1 Integration

The functionality provided by the ADG to the SOCCRATES platform is utilized via the ADG Cortex analyser and its main function “simulation”. The simulation function exposed by the Cortex analyser

[8] https://github.com/foreseeti/securicad-enterprise-sdk

handles all Use Cases provided by the ADG and takes a set of optional parameters depending on which Use Case in the workflow that has been triggered. All parameters are optional and the ADG assigns default values for all omitted parameters. The core functionality provided via the parameters are:

- **Date** (date)
  The ADG can perform simulations on historical data for comparisons over time or post-mortems. The date parameter specifies from which timestamp to collect the model data from the IMC. (Use Case 4 & 5)

- **Threat actor** (threatActor)
  The simulations performed depends on the capabilities of the simulated attacker. These capabilities can be fetched from the TIP based on a Threat actor name or by configuring your own attacker by providing a list of capabilities. (Use Case 2)

- **Attacker entry points** (entryPoints)
  The attacker entry points decides where the attacker will start in the simulation. The entry point can represent an external attacker that starts on the Internet or to represent an ongoing attack or IoC. The entry points is a list of assets and types of attacks that the attacker will perform at the start of the simulation. (Use Case 1 & 2)

- **High value assets** (highValueAssets)
  The simulated attacker will try to reach the assets marked as high value assets in the model and the results will be focused around these assets. (Use Case 1 & 2)

- **Vulnerabilities** (vulnerabilities)
  By default, all vulnerabilities reported by the IMC will be included in the simulation and considered as a potential exploit for the attacker. However, to perform an impact analysis of newly discovered vulnerabilities for a set of specific assets or vulnerabilities, these can be provided as a parameter. All other vulnerabilities will be omitted from the simulation. (Use Case 3)

#### 3.4.2.2 Integration configuration

The data model described in the next section requires configuration of the ADG Cortex analyse. The configuration items are listed in Table 3.7.

**Table 3.7 - ADG Cortex analyser configuration items**

| ADG Cortex analyser configuration items |
|---|
| **simulation**<br>Start a simulation in the ADG |
| **Configuration items** |
| **url** (required) string<br>URL of the securiCAD service<br><br>**username** (required) string<br>Username of the securiCAD user<br><br>**password** (required) string |

| |
|---|
| Password of the securiCAD user<br><br>**organization_name** (required) string<br>Organization of the securiCAD user<br><br>**project_name** (optional) string<br>Name of the Project to simulate in. If empty, a new one will be created.<br><br>**scenario_name** (optional) string<br>Name of the Scenario to simulate in. If empty, a new one will be created.<br><br>**imc_url** (required) string<br>URL of the IMC component<br><br>**imc_username** (required) string<br>Username of the IMC user<br><br>**imc_password** (required) string<br>Password of the IMC user<br><br>**tip_url** (required) string<br>URL of the TIP component<br><br>**tip_username** (required) string<br>Username of the TIP user<br><br>**tip_password** (required) string<br>Password of the TIP user |

### 3.4.3 Data model

The table below describes the parameters available when starting a simulation in the ADG through the ADG Cortex analyser.

**Table 3.8 - ADG Cortex analyser request and response parameters**

| ADG Cortex analyser functions |
|---|
| **simulation**<br>Start a simulation in the ADG |
| **Request parameters** |
| **date** (optional) string<br>The date from which securiCAD will fetch a snapshot from the IMC. Default: Latest snapshot.<br>Example: "2020-10-20"<br><br>**threatActor** (optional) object<br>The threat actor name or MITRE ATT&CK group ID (e.g. “G0007”) from the TIP to simulate with and/or a custom attacker profile from a list of capabilities (MITRE ATT&CK technique IDs). Default: Generic attacker.<br>Example:<br>`{`<br>`"threatActor": "APT-28",` |

```
    "capabilities": ["T1110.002", "T1087"]
}
```

**entryPoints** (optional) array
An array of objects representing attacker entry points given current e.g. IoCs, with a reference "ref" to an asset ID in the IMC and type of attack. Default: Internet attacker.
Example:

```
[
    {
        "ref": "112233",
        "type": "Phishing"
    }
]
```

**highValueAssets** (optional) array
An array of objects representing attacker targets given current e.g. IoCs, with a reference "ref" to an asset ID in the IMC and type of impact. Default: High value assets according to the IMC.
Example:

```
[
    {
        "ref": "112233",
        "type": "Confidentiality"
    }
]
```

**vulnerabilities** (optional) array
An array of objects representing a vulnerability "cve" with a reference "ref" to an asset ID of a Host in the IMC with the vulnerability to include in the simulation. Default: All vulnerabilities in the current IMC snapshot.
Example:

```
[
    {
        "ref": "112233",
        "cve": "CVE-2021-1234"
    }
]
```

**Response (Success)**

**simulationId** string
A unique ID for the simulation
Example: "123456789123"

**reportUrl** string
URL to the report in securiCAD
Example: "https://10.0.0.2/org/project/123456789123"

**ciaResults** array
An array of objects with a reference "ref" to an asset ID in the IMC and the TTC and probability of C/I/A compromise.
Example:

```
[
    {
        "ref": "112233"
        "confidentiality": {
            "ttc": 1,
            "probability": 1.0
```

```
        },
        "integrity": {
            "ttc": 10,
            "probability": 0.5
        },
        "availability": {
            "ttc": 3,
            "probability": 0.8
        }
    }
]
```

**ciaResultsDelta** array

An array of objects with a reference "ref" to an asset ID in the IMC and the delta TTC and probability of C/I/A compromise compared to the previous simulation.

Example:

```
[
    {
        "ref": "112233"
        "confidentiality": {
            "ttc": -5,
            "probability": 0
        },
        "integrity": {
            "ttc": 0,
            "probability": -0.3
        },
        "availability": {
            "ttc": 1,
            "probability": 0
        }
    }
]
```

**results** object

An object with all results from the simulation in the ADG

Example:

```
{
   "model": {"objects": [], "associations": []},
   "simid": "337392601332156"
   "results": {"risks": [{"id": "", "ttc": 10}], "risk": 87},
   "threat_summary": [{
        "attackstep": "CodeExecution",
        "id": "",
        "user": "Vulnerabilities in the application could allow …"
        "stride": "Tampering",
        "mitre": "T1210 Exploitation of Remote Services"
   }],
   "suggested_mitigations": […],
   "critical_paths": [{
        "nodes": [
            {"name": "", "id": "", "attackstep": ""}
        ],
        "links": [
            {"source": "<id>", "target": "<id>"}
        ],
        "target": {"name": "", "id": "", "attackstep": ""}
   }],
   "report_url": "https://10.0.0.2/org/project/123456789123"
}
```

**Response (Failure)**

| **errorMessage** string<br>Error output message from the ADG. |
|---|

## 3.5 Course of Action Generator

The Course of Action (CoA) generator operates on a model of an IT architecture as used by the ADG and generates a set of defences that are inactive in the model and are suggested to activate.

### 3.5.1 Functional specification

The fundamental function of the CoA generator is to suggest defences improving the security of ICT architecture (as captured in the model of an IT architecture used by the ADG) to the SOC analyst.

Functions of CoA generator:
Input:

- Mandatory: Pointer to an attack defence graph model in the form of securiCAD simulation id. In this model one or more target attack steps are specified representing high value assets that are the ones that are to be protected. (To be provided by the ADG component.)
- Optional: The information on cost of implementation of defences (potentially in multiple cost dimensions) and the available budget. (To be provided by the SOC analyst.)

Output:

- Initial Time-to-compromise values for each target attack step in the securiCAD model
- A list of CoAs, each CoA includes the following information
    - Efficiency score of the defences as calculated by the CoA generator used for prioritization inside the CoA generator algorithm
    - Projected time-to-compromise values after implementing the defences in the course of action
    - Cost of implementation of CoA (in multiple cost dimensions)
    - Set of ordered defences in the securiCAD model from the ADG component
    - Web Link to the simulation report in securiCAD after adding the CoA defences

The CoA generator forms a feedback loop with the ADG analyser. To drive the optimization process, the generator computes a partial response to the threat based on the input attack path, run the ADG analyser to obtain the expected behaviour of the attacker under this partial response, repeat the procedure on the newly obtained attack path, etc. These iterations thus happen *inside* the CoA generator component without the involvement of the OIE.
By altering different budgets (e.g. time and cost) different kinds of CoAs can be computed as per the requirement viz. quick response CoA, strategic CoA, etc.

### 3.5.2 API specification

CoA Cortex Analyser is written to integrate CoA generator to the OIE of SOCCRATES. Simulation id of an ADG analysis is required by CoA generator to start its computation, there are a few optional request parameters (costs, budgets, and inherent design restrictions). Figure 3.7 depicts the integration architecture for the CoA generator.

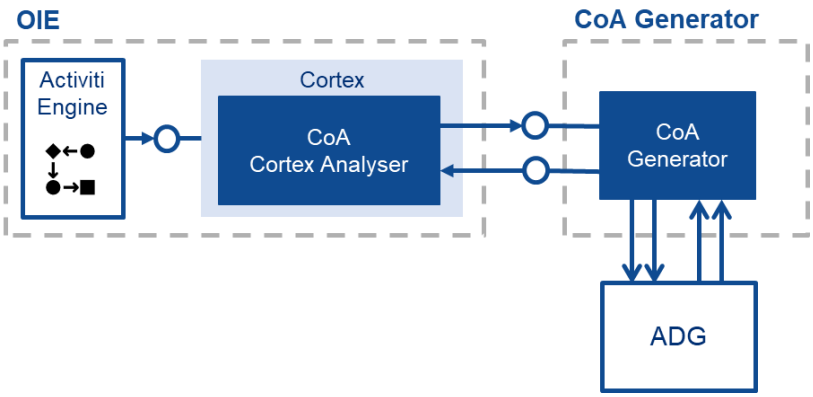


**Figure 3.7 – Integration architecture for CoA Generator**

### 3.5.2.1 Integration

The functionality provided by the CoA to the SOCCRATES platform is utilized via the CoA Cortex Analyser. The simulation id is a mandatory parameter required whereas all other parameters are optional. If budgets are not provided, then CoA generation will assume infinite budget. If costs are not provided zero costs will be assumed. If no design restrictions are provided none will be assumed. The core functionality provided via the parameters are:

- **Simuation id**
  - Pointer to an attack defence graph (in the form of SecuriCAD simulation id) generated from the existing IT infrastructure for target attack steps of high value assets.
- **Monetary Budget(s) (Optional)**
  - Values specifying available (monetary) resources of different kinds
    e.g., budget for capital expenditures and budget for operating expenditures.
    An infinite number of budgets can be defined.
- **Time Budget (Optional)**
  - Available (time-like) resources.

The costs related to the implementation of defences will have to be defined. The current idea is to include such cost pre-defined with the definition of Defenses[9] in the ADG, but this is for further study as these costs may depend on many factors. The costs related to the implementation of defences can be expressed with the following parameters:

- **Monetary Cost for each Defense (Optional)**
  - According to the specified budgets, respectively (e.g. capex and opex)
  - Costs can be expressed as depending the number of times a defence type is implemented/instantiated. (e.g. for marginal cost. Applying an IDS the first time is costly while implementing a second clone instance of the IDS is much cheaper.)
- **Time Cost for each Defense (Optional)**
  - Costs can be expressed as depending the number of times a defence type is implemented/instantiated.
- **List of Pre-requisites for a particular Defense (Optional)**

[9] In securiCAD, the basis for the ADG, the US spelling for defence is used. In this report, we therefore use the US spelling for defence were it specifically refers to Defense in secuiCAD.

- All of the defences in prerequisites need to be implemented before the considered defence can be implemented

- **List of Exclusive Defenses (Optional)**
  - Only one defence in an exclusive cluster can be implemented in the system

#### 3.5.2.2 Integration configuration

The configuration items listed in Table 3.9 are required in the CoA Cortex analyser to facilitate the communication between the CoA generator and the ADG analyser.

**Table 3.9 - CoA Cortex analyser configuration items**

| CoA Cortex analyser configuration items |
|---|
| communication with ADG analyser |
| **Configuration items** |
| **scad_url** (required) string<br>URL of the securiCAD service<br><br>**username** (required) string<br>Username of the securiCAD user<br><br>**password** (required) string<br>Password of the securiCAD user<br><br>**organization_name** (required) string<br>Organization of the securiCAD user<br><br>**project_name** (required) string<br>Name of the Project in securiCAD |

### 3.5.3 Data model

Parameters required with the CoA generation request are given in Table 3.10.

**Table 3.10 - CoA generation request and response parameters.**

| CoA Cortex Analyser functions |
|---|
| **coagen**<br>generate CoA for the model provided |
| **Request parameters** |
| **simulationId** string<br>Example:<br>`"simulationId" : "123456789123"`<br><br>**monetary_budgets** (optional) Dictionary.<br>Values specifying available (monetary) resources, e.g., budget for capital expenditures and budget for operating expenditures. It is a dictionary with keys being consecutive numbers starting with 1, as strings, and with values being floats. These keys represent different monetary budget resources. For instance, we can call the capital cost budget "1" and operational cost budget "2".<br><br>Example: |

```
"monetary_budgets": {
   "1": value,
   …
   "n": value
}
```

**time_budgets** (optional) Dictionary
Float values specifying available (time-like) resources. In practice, one value is expected, expressing a limit on time available for implementation of a CoA. It is a dictionary with keys being consecutive numbers starting with 1, as strings, and with values being floats. Similar to monetary budgets we can have multiple time budget resources.
Example:

```
"time_budgets": {
   "1": value,
   …
   "n": value
}
```

**Response (Success)**

**Initial ttc values.** Dictionary
Initial time-to-compromise values (ttc5, ttc50, and ttc95) for each target attack step in securiCAD model.
Example:

```
   "initialTTC": {
      "228.Read": [
        19,
        26,
        34
      ],
      "325.NetworkRespondConnect": [
        23,
        30,
        40
      ]
    }
```

**CoAs.** array
List of course of actions (CoA). Each CoA contains 1) a value for quantifying the quality of the CoA "efficiency", 2) projected time-to-compromise values after implementing the defences in CoA, 3) time and monetary costs for implementing CoA, 4) a list of "*defenses*" to be applied, and finally 5) URL to the simulation report in securiCAD after adding the CoA defences “report_url”. Cost dimensions will follow different budget resources provided as request parameter. Each defense has a reference "ref" to an asset ID in the securiCAD (ADG), a defence in securiCAD "defenseName", a description of the defence "defenseInfo" and a reference to a mitigation in mitre attack "mitreRef".
Example:

```
    [
      {
        "efficiency": "170.585",
        "coaTTC": {
          "228.Read": [
            18,
            25,
            34
          ],
          "325.NetworkRespondConnect": [
            1.7976931348623157e+308,
```

```
            1.7976931348623157e+308,
            1.7976931348623157e+308
          ]
        },
        "time_cost": {
          "1": value,
          …
          "n": value
        },
        "monetary_cost": {
          "1": value,
          …
          "n": value
        },
        "defenses": [
          {
            "ref": "100",
            "defensename": "disabled",
            "defenseInfo": "Disable Application",
            "mitreRef": "M1042"
          }
        ]
         "report_url":"https://10.0.0.2/org/project/123456789123/sce-
nario/232236696159292/report/112654926483686"
      },
      {
        "efficiency": "272.808",
        "coaTTC": {
          "228.Read": [
            1.7976931348623157e+308,
            1.7976931348623157e+308,
            1.7976931348623157e+308
          ],
          "325.NetworkRespondConnect": [
            1.7976931348623157e+308,
            1.7976931348623157e+308,
            1.7976931348623157e+308
          ]
        },
        "time_cost": {
          "1": value,
          …
          "n": value
        },
        "monetary_cost": {
          "1": value,
          …
          "n": value
        },
        "defenses": [
          {
            "ref": "119",
            "defensename": "authenticated",
            "defenseInfo": "Two Factor Authentication (2FA)",
            "mitreRef": "M1032"
          },
          {
            "ref": "100",
            "defensename": "Disabled",
            "defenseInfo": "Disable Application",
            "mitreRef": "M1042"
          }
```

```
        ]
        "report_url":"https://10.0.0.2/org/project/123456789123/sce-
nario/232236696159292/
report/337386199836050"
      }
    ]
```

**Response (Failure)**

**errorMessage** string
Error output message from the CoA.

## 3.6 Business Impact Analyser & Business Logic Modelling

### 3.6.1 Functional specification

The Business Impact Analyser (BIA) is a component that aims at providing impact assessment on business functions and processes. It uses the output of the ADG analyser as a starting point to assess the business impact in case an attacker successfully compromises the target node at the end of the attack path. Therefore, the BIA does not reason in terms of attack path, but in terms of impact propagation to other assets, and ultimately to business functions and processes. To do so, the BIA relies on Business Logic Modelling (BLM), which builds a graph model of assets and business entities, as described in detail in deliverable D5.1 [10].

The list of assets can be retrieved from the IMC, and the dependencies between assets are derived from Netflow data, which are updated every 24 hours. The graph of the dependencies between business entities must be established manually beforehand by the pilot sites, and will only be updated in Use Case 5.

Once the current model is generated, the BIA is able to provide computation of business impact given an external event (attack or CoA deployment) affecting one or several assets. The BIA is also able to compute a criticality value for all the assets, which makes it possible to quickly get an overview of the most critical assets, in terms of how much these assets contribute to propagate impact up to the most important business entity.

Upon analysing the impact of an external event, the BIA provides two outputs:

- The impact probability on the most important business entity
- The critical time, under which no significant impact on the company is to be expected

In the case where the external event is an attack, these outputs will be transmitted to the SOC analyst in order to assess the level of emergency of the situation. In the case where the external event is the deployment of a CoA, the outputs will be added to the results of the Response Planner to help the SOC analyst decide which CoA should be deployed in priority.

### 3.6.2 API specification

The BIA has a REST API that can be used to request the impact analysis of an event, or to query the critical assets of the infrastructure. In the context of the SOCCRATES platform, the BIA is always

triggered by the OIE. Therefore, the interactions with the API go through Cortex analysers or Cortex responders. More specifically, 2 Cortex analysers are defined for all the use cases to send the queries to the BIA API, and 1 Cortex responder is defined for use case 5 in order to update the model built by the BLM. Figure 3.8 depicts the integration architecture for the BIA component using Cortex.

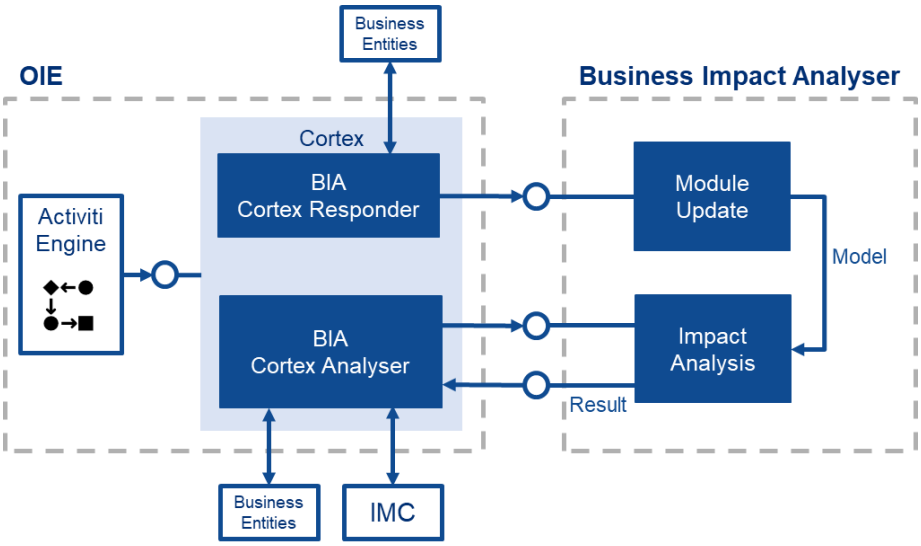


**Figure 3.8 – Integration architecture for BIA**

The model on which the business analysis is performed is built based on two data sources: Netflow data obtained from the IMC, and the graph of business entities obtained as a configuration file. The latter is how this will be done in the pilot sites. It is for further study how to automatically collect such data from an API to for instance an Enterprise Architecture product. For Use Case 5, the model update will be triggered upon the addition of a new system in the infrastructure, which requires an update of the graph of business entities on the pilot sites.

## 3.6.3 Data model

### 3.6.3.1 Cortex analysers

The queries to the BIA can be sent through two functions, whose specification is described in the table below. One cortex analyser must be specified for each function.

**Table 3.11 – BIA cortex analyser request and response parameters for bia_analysis**

| **Business Impact Analyser** | |
|---|---|
| **Function in the cortex analyser** | *bia_analysis* |
| **Request Data** | |
| **Impacted assets** Array<br>In order to compute the business impact, it is necessary to know which assets are (potentially) impacted. This can be based he results from the SIEM or output of the ADG analyser. For each impacted asset, it is also necessary to know the level of compromise in terms of confidentiality, integrity and availability, based on the output of the ADG analyser.<br>Example in the case of a single asset:<br>`[`<br>`  {`<br>`  "assetID": "112233"` | |

```
        "confidentiality": {
            "ttc": 1,
            "probability": 1.0
        },
        "integrity": {
            "ttc": 10,
            "probability": 0.5
        },
        "availability": {
            "ttc": 3,
            "probability": 0.8
        }
    }
]
```

**Business function/process name** (optional) Array of strings
The business impact can be computed on any existing business function or process. If no input is provided, the business impact will be computed for the highest-level business entity (named « Business Company »).

```
business_name: ["value_1",..., "value_n"]
```

**Attack time** (optional)
The attack time can be useful for the computation of the impact of attacks that may occur at a certain time, possibly different from the current time. If no input is provided, the computation will be performed for the current time.

```
attack_time: value
```

**Response Data**

**Business impacts** Array
The business impact computed for each business function/process provided in the input. The business impact is expressed through three values: the impact probability, impact criticality (value between 0 and 10), and critical time expressed in hours. The impact probability represents the probability that a business process or function cannot work as intended, the impact criticality represents the importance of the business process / function for the company, and the critical time represents the time under which the business process / function will suffer no significant impact.

**Report URL** String
The URL allowing the SOC analyst to access the GUI of the BIA and display the full graph corresponding to the request processing.

Example:

```
{
  business_impacts:
 [
    {
        "business_name": "Business Company"
        "impact_probability”: 0.67
        "impact_criticality": 6
        "critical_time": 2.4
    }
```

```
  ]
  report_url: "https://10.0.0.2/org/project/123456789"
}
```

**Table 3.12 – BIA cortex analyser request and response parameters for bia_fetchCriticalAssets**

| **Business Impact Analyser** | |
|---|---|
| **Function in the cortex analyser** | *bia_fetchCriticalAssets* |
| **Request Data** | |
| **Threshold** (optional) Float<br>The criticality value above which an asset will be considered critical. The value must be greater than 0 and lower than 1. The default value is 0.66.<br><br>`threshold: value` | |
| **Response Data** | |
| **Critical assets** Array of strings<br>The output is the list of critical assets, i.e. assets that contribute the most to the propagation of impact towards business entities. The assets are identified by the IMC identifier.<br><br>`critical_assets: ["value_1",..., "value_n"]`<br><br>**Report URL**<br>The URL allowing the SOC analyst to access the GUI of the BIA and display the full graph corresponding to the request processing.<br><br>`report_url: "https://10.0.0.2/org/project/123456789"` | |

### 3.6.3.2 Cortex responder

**Table 3.13 – BIA cortex responder request parameters.**

| **Business Impact Analyser** | |
|---|---|
| **Function in the cortex responder** | *bia_updateModel* |
| **Request Data** | |
| **newAsset** String<br><br>The IMC identifier of the new asset that was added to the infrastructure. | |
| **Response Data** | |
| **None** | |

## 3.7 Response Planner

### 3.7.1 Functional specification

The Response Planner (RP) has two main functions in the SOCCRATES platform:

1) for all use cases, the RP will estimate the financial and operational impact of CoAs obtained from the CoA generator, and rank these CoAs based on the Return On Response Investment (RORI) metric.
2) the RP also generates containment CoAs to mitigate an ongoing attack in use case 1.

The RORI metric provides a quantification of the financial impact of a CoA. It can be either positive, meaning that the deployment of the CoA is beneficial, or negative, which means that the CoA should not be deployed at all. It is computed by taking into account various parameters, such as the financial impact of the current attack on the company, the size of the company, the cost and collateral damage of the CoA, and the risk mitigation provided by the CoA. Therefore, it is possible to compare various CoAs based on this metric and on the operational impact computed by the BIA. These values are then presented to the SOC analyst as a support for decision making.

In the context of use case 1, the CoA generator from the SOCCRATES platform is not optimised for generating CoAs that are aimed at stopping the ongoing attack. Therefore, the RP will generate containment CoAs whose main purpose will be to isolate compromised systems and prevent data exfiltration. These containment CoAs will be generated based on SIEM events. In the same way as the other CoAs, the RORI value and the operational impact will be computed for each containment CoA, to assist the SOC analyst in decision making.

The output of the RP will be a list of CoAs, with for each CoA the financial and operational impact values. Each CoA will be represented as a CACAO playbook along with a human readable label.

### 3.7.2 API specification

The RP has a REST API that can be used to request the computation of the RORI value of various CoAs, as well as to trigger the generation of containment CoAs for use case 1. In the context of the SOCCRATES platform, the BIA is always triggered by the OIE. Therefore, the interactions with the API go through 2 Cortex analysers: one dedicated to the RORI computation, and one dedicated to the containment CoA generation. Figure 3.9 depicts the integration architecture for the RP component using Cortex.

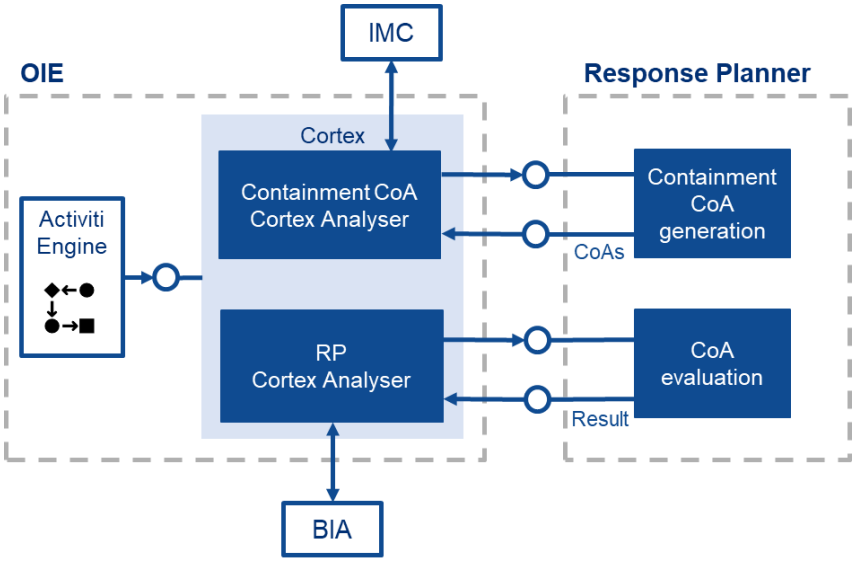


**Figure 3.9 – Integration architecture for RP**

The generation of the containment CoAs will rely on the IMC to retrieve information about the security controls. Regarding CoA evaluation, the RP will need to send a request to the BIA to obtain an evaluation of the business impact of the CoAs.

### 3.7.3 Data model

**Table 3.14 – RP cortex analyser request and response parameters for coa_ranking**

| **Response Planner Cortex analyser** | |
|---|---|
| **Function in the cortex analyser** | coa_ranking |
| **Request Data** | |

**siem_label (optional)** String
The type of event for which CoAs are generated. Possible values are: "TA0001" (Initial access), "TA0008" (Lateral movement), or "TA0010" (Exfiltration).

**initialTTC (optional)** Dictionary
Initial time-to-compromise values (ttc5, ttc50, and ttc95) for each target attack step in secu-riCAD model.

**containment (optional)** Boolean
This value indicates whether the CoAs in the current input are containment CoAs or not. Default value is false.

**coas** Array
The CoAs are obtained either from the CoA generator or from the containment CoA generator (which is part of the RP). The RORI value will be computed for each of the CoAs received in the input. The new TTC value as well as the cost associated to each CoA must be provided. In the case of a containment CoA, only the "defenses" part is expected.

Example:

```
{
  siem_label:”TA0001”,
  initialTTC: {
      "228.Read": [
        19,
        26,
        34
      ],
      "325.NetworkRespondConnect": [
        23,
        30,
        40
      ]
   }
  containment: false,
  coas:
  [
      {
        "efficiency": "170.585",
        "coaTTC": {
          "228.Read": [
```

```
            18,
            25,
            34
          ],
          "325.NetworkRespondConnect": [
            1.7976931348623157e+308,
            1.7976931348623157e+308,
            1.7976931348623157e+308
          ]
        },
        "time_cost": {
          "1": value,
          …
          "n": value
        },
        "monetary_cost": {
          "1": value,
          …
          "n": value
        },
        "defenses": [
          {
            "ref": "100",
            "defensename": "Disabled",
            "defenseInfo": "Info",
            "mitreRef": "Ref"
          }
        ]
      },
      {
        "efficiency": "272.808",
        "coaTTC": {
          "228.Read": [
            1.7976931348623157e+308,
            1.7976931348623157e+308,
            1.7976931348623157e+308
          ],
          "325.NetworkRespondConnect": [
            1.7976931348623157e+308,
            1.7976931348623157e+308,
            1.7976931348623157e+308
          ]
        },
        "time_cost": {
          "1": value,
          …
          "n": value
        },
        "monetary_cost": {
          "1": value,
          …
          "n": value
        },
        "defenses": [
          {
            "ref": "119",
            "defensename": "authenticated",
            "defenseInfo": "Info",
            "mitreRef": "Ref"
          },
          {
            "ref": "100",
```

```
            "defensename": "Disabled",
            "defenseInfo": "Info",
            "mitreRef": "Ref"
          }
        ]
      }
    ]
}
```


**Response Data**

**Ranked coas** Array

The output is an array of CoAs of the same length as the one provided as input, providing the description of the CoAs as well as, in the case of CoAs that are not containment CoAs, their RORI value and their business impact, ranked by highest RORI value. In the case of containment CoAs, no RORI value is computed, therefore the CoAs are ranked by lowest business impact. RORI and Business impact are of type Float. The CoAs are expressed as a CACAO playbook.

**Report URL** String

The URL allowing the SOC analyst to access the GUI of the RP and display the full graph corresponding to the request processing.

Example with one evaluated CoA:


```
{
  ranked_coas:
  [
    {
        "containment": false,
        "RORI": 76,
        "business_impact": 0.56,
        "playbook":
       {
        "type": "playbook",
        "spec_version": "1.0",
        "id": "playbook--soccrates-coa-123456123456",
        "name": "Soccrates CoA Playbook",
        "description": "This playbook will apply the CoA in the system",
        "playbook_types": [
             "prevention"
         ],
        "created_by": "identity--uuid",
        "created": "2021-07-19T23:32:24.399Z",
        "labels": [
           "Patch",
           "2FA",
           "CoA"
        ],
        "features": {
           "parallel_processing": true,
        },
        "workflow_start": "start--soccrates-coa-123456123456",
        "workflow":
       {
          "start--soccrates-coa-123456123456": {
          "type": "start",
          "on_completion": "parallel--soccrates-coa-123456123456"
         },
```

```
        "parallel-soccrates-coa-123456123456": {
          "type": "parallel",
          "name": "Apply defenses of the Coa",
          "next_steps": [
            "defense1--soccrates-coa-123456123456",
            "defense2--soccrates-coa-123456123456"
          ]
        },
        "defense1--soccrates-coa-123456123456": {
          "type": "single",
          "name": "Patch Application",
          "description": "This step will Patch the application on selected
hosts",
          "on_completion": "end--soccrates-coa-123456123456",
          "target": "112233",
          "commands": [
            {
            "type": "manual",
            "command": "Download and install the patch for the application in
selected hosts"
            }
          ]
        },
        "defense2--soccrates-coa-123456123456": {
          "type": "single",
          "name": "Two Factor Authentication",
          "description": "This step will enable Two Factor Authentication",
          "on_completion": "end--soccrates-coa-123456123456",
          "target": "445566",
          "commands": [
           {
            "type": "manual",
            "command": "Enable Two Factor Authentication on the selected hosts"
           }
          ]
         },
       "end--soccrates-coa-123456123456": {
       "type": "end"
       }
      }
     }
   }
 ]
 report_url: "https://10.0.0.2/org/project/1234567890"
}
```


**Table 3.15 – RP cortex analyser request and response parameters for generateContainmentCoas**

| **Containment CoA Generator Cortex analyser** | |
|---|---|
| **Function in the cortex analyser** | generateContainmentCoas |
| **Request Data** | |
| **siem_label (optional)** String<br>The type of event provided by the SIEM. Possible values are: "TA0001" (Initial access), "TA0008" (Lateral movement), or "TA0010" (Exfiltration)<br><br>**sourceIPs** list of Strings<br>List of source IP addresses identified in the SIEM event. At least one source IP address is expected | |

**destinationIPs** list of Strings
List of destination IP addresses identified in the SIEM event.

Example:

```
{
   siem_label: "TA0001",
   sourceIPs: [
    "192.168.0.3",
    "192.168.0.10"
  ],
   destinationIPs: [
    "123.45.67.89",
    "123.98.76.54"
  ]
}
```

**Response Data**

**containment** Boolean
This value indicates whether the CoAs are containment CoAs or not. The default value will be true.

**coas** Array

The output is an array of CoAs.
Example with one CoA:

```
[
    "containment": true
    "coas":
    {
        "defenses": [
            {
                "ref": "112233",
                "defenseName": "Isolate",
                "defenseInfo": "Network Segmentation",
                "mitreRef": "M1030",
                "hosts": ["192.168.0.3", "192.168.0.10"]
            },
            {
                "ref": "112233",
                "defenseName": "Filter",
                "defenseInfo": "Filter Network Traffic",
                "mitreRef": "M1037"
                "flows": [
                  {
                   "src": "192.168.0.3",
                   "dst": "123.45.67.89"
                  }
                 ]
            }
        ]
    }
]
```

## 3.8 Threat Intelligence Platform

The Threat Intelligence Platform (TIP), based on the open-source ACT platform developed by mnemonic, provides contextualised threat intelligence to the SOCCRATES platform.

### 3.8.1 Functional specification

The Threat Intelligence Platform (TIP) continuously collects threat intelligence from a diverse set of sources, enriches and correlates the intelligence, and provides interfaces for accessing the intelligence. The primary roles of the TIP in the SOCCRATES platform are to provide related IoCs for UC1, and to trigger UC2 when new relevant threat intelligence is obtained.

### 3.8.2 API specification

The ACT platform provides a set of versioned REST APIs in order to interact with the system. A Python library[10] to interact with the API is available, as well as a graphical user interface[11] (GUI). The API endpoints are documented with Swagger, which contains a full description of each endpoint, including requests and responses. The Swagger documentation is available at https://act-eu1.mnemonic.no/swagger/.

Since the OIE does not directly interact with the TIP from the workflows, no Cortex analyser for the TIP has been developed.

#### 3.8.2.1 Authentication

Secure authentication currently requires a reverse proxy in front of the REST API that can map authenticated users to a corresponding ACT-User-ID. The ACT-User-ID must then be sent with a special HTTP header with every request. It identifies the user and contains the user's numeric ID as defined in the access controller's properties file. Sending a different header value means executing the request as a different user.

#### 3.8.2.2 Creating ObjectTypes and FactTypes

The API allows for the creation of ObjectTypes (such as 'ip' and 'fqdn') and FactTypes which describes the relationship between two ObjectTypes (for instance the relationship between ip and fqdn).

**Table 3.16 – API for the creation of ObjectTypes**

| /v1/objectType | Creation of ObjectTypes |
|---|---|
| Method | POST |
| Parameters | • name<br>• validator |
| Response | UUID |

**Table 3.17 – API for the creation of FactTypes**

| /v1/factType | Creation of FactTypes |
|---|---|
| Method | POST |
| Parameters | • name<br>• validator |

[10] https://github.com/mnemonic-no/act-api-python

[11] https://github.com/mnemonic-no/act-frontend

| | |
|---|---|
| | • relevantObjectBindings |
| Response | - |

**Table 3.18 – API to add a Fact**

| /v1/fact | Creation of Facts |
|---|---|
| Method | POST |
| Parameters | • type<br>• sourceObject<br>• destinationObject<br>• bidirectionalBinding |
| Response | |

When sending the same “add” request multiple times, only the lastSeenTimestamp in the response will be updated. The platform ensures that the same Facts are not added multiple times, it will just record that the Fact has been seen again.

Every Fact is owned by an organisation and access to Facts is strictly controlled. A Fact can be public such that every user can see it. A Fact can have explicit access control which means that a user must be explicitly granted access to that Fact. The default access mode is role-based. In this mode a user can see the Fact if one has the general permission to view Facts for the organisation owning the Fact, or if one was granted explicit access.

**Table 3.19 – API to search through the stored facts**

| /v1/fact/search | Search through facts |
|---|---|
| Method | POST |
| Parameters | keywords |
| Response | Fact(s) |

### 3.8.2.3 Manipulating Facts

There are multiple endpoints which operate on Facts besides adding and fetching Facts. It is possible to add comments and retrieve them, to grant users access to non-public Facts, and to retract Facts. Retracting a Fact will add another Fact to the platform which states that the retracted Fact is not true. It acts as a revocation.
Furthermore, the platform supports meta Facts - i.e. Facts about Facts. A meta Fact is a Fact directly bound to another Fact without the link via an Object, for instance a timestamp or Organization identifier related to a fact. It adds more information directly to a Fact. Specific endpoints exist for adding and retrieving meta Facts. Available meta Facts must be declared beforehand with FactTypes like the resolve FactType has been defined above.

### 3.8.2.4 Manipulating Objects

Objects cannot be added manually to the platform because they conceptually always "exist". They are stored automatically the first time a Fact linking to them is added. Therefore, it is only possible to retrieve Objects. They can either be fetched by UUID - GET /v1/object/uuid/{id} - or by type and value - GET /v1/object/{type}/{value}.

**Table 3.20 – API to retrieve object by UUID**

| /v1/object/uuid/{id} | Retrieve Object |
|---|---|
| Method | GET |
| Parameters | UUID |
| Response | • Object<br>• Statistics about all Facts bound to the Object |

**Table 3.21 – API to retrieve Object by type and value**

| /v1/object/{type}/{value} | Retrieve Object |
|---|---|
| Method | GET |
| Parameters | • type<br>• value |
| Response | • Object<br>• Statistics about all Facts bound to the Object |

One very powerful functionality is implemented in the /v1/traverse endpoints. These endpoints allow to query the information stored in the platform using the Gremlin query language provided by Apache TinkerPop. The API for this is shown in Table 3.22. The query starts at an FQDN Object and moves via Facts to corresponding IPv4 Objects. Although this is an easy example, graph queries are very powerful and allow the analyst to uncover previously unknown information which was not observable by just looking at the raw data.

**Table 3.22 – API to query using Gremlin graph traversal**

| /v1/traverse/object/{type}/{value} | Retrieve results of graph traversal starting at specified object |
|---|---|
| Method | POST |
| Parameters | • type (e.g. fqdn)<br>• value (e.g. www.github.com) |
| POST parameters | • query (e.g. g.out('resolvesTo')) |
| Response | • Results of graph traversal (objects and/or facts) |

### 3.8.3 Data model

Figure 3.10 shows the current TIP data model. The nodes represent object types, while the edges represent fact types, as described in D2.2 Section 10.2 [2].

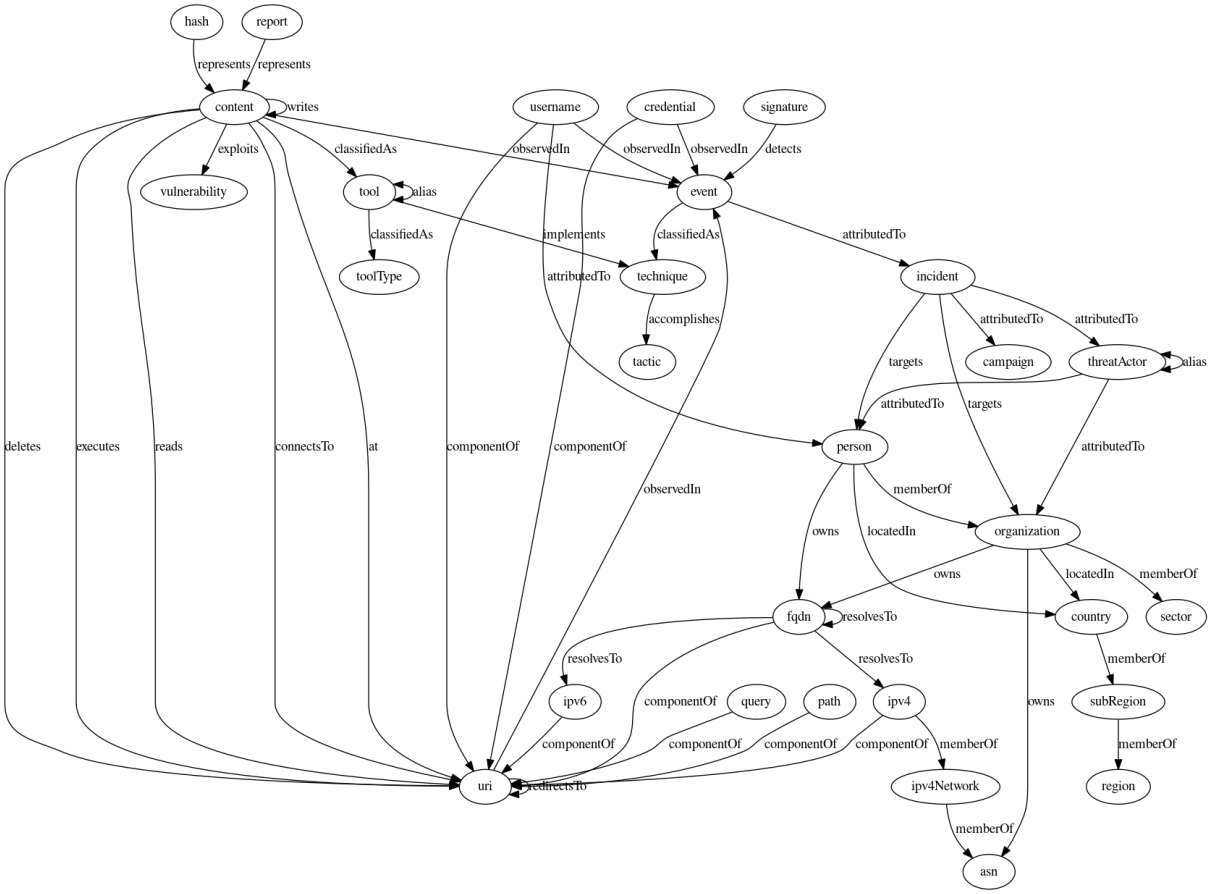


**Figure 3.10 – TIP data model**

The TIP has functionality that enables triggering of the Orchestration & Integration engine when new Facts that match predefined criteria are added to the graph (UC2). Furthermore, the graph can be queried using the REST API and/or Python library, and the TIP supports graph queries using the Gremlin query language[12], as shown in Table 3.22 (UC1).

Example:
One of the scenarios for UC2 is that the TIP receives information that exploit code for a vulnerability is available. In the TIP graph, this information is encoded as an **exploits** edge that connects a **content** object to a **vulnerability** object. A TIP rule that triggers a REST API call to the Orchestration & Integration Engine when a new **exploits** edge is added to the graph will trigger UC2.

## 3.9 Threat Data Collection & Threat Prediction

This section updates the initial Threat Data Collection & Threat Prediction interface specification in D2.2 [2]. It specifies API methods implemented at Shadowserver that enable access to machine learning sourced knowledge derived from the analysis of malware runs in the Shadowserver Sandbox (T4.2). It also provides access to trends observed in the Sandbox DNS queries, as well as training datasets that can be used to build ML based DGA detection models. The API can be used by other SOCCRATES components, notably the TIP (T4.4) and other AI Based Attack Detection Components (developed in Task 4.1).

[12] https://tinkerpop.apache.org/gremlin.html

The API is a live component, which we expect may still evolve till the end of the project to provide better functionality and new features, should any be needed.

### 3.9.1 Functional specification

The component will provide the following functionality:

1. Query for a domain name and provide a classification of whether or not the domain is likely to be used in a DGA, and optionally the likely associated malware family.
2. Query for a malware hash and provide any associated DGA domains discovered through the AI based methods developed.
3. Query for the latest DGA domains tagged by AI based methods developed
4. Query for the latest DGA training set data for the anomaly detection tools developed in task 4.1.
5. Query for the latest TOP domain lists, which can also be used as a training set with (potentially) benign data for the anomaly detection tools developed in task 4.1
6. Query for latest trends seen in DNS traffic, for example, by TLD, crypto domains or registrar.

### 3.9.2 API specification

The API requires that clients send their API key as well as a hash-based message authentication code (HMAC) of the request data created with a secret given to each client. Recalculating and comparing the hash allows the server to authenticate the user and ensure that the request has not been tampered with.

The API is mostly RESTful with the following stipulations:

- API calls must be made with HTTP POST.
- All requests are JSON objects.
- Each request requires an API key and HMAC header.
- Any non-200 HTTP response is an error. Error details will be included in the response object.

The following API methods specified below offer the functionality described in 3.9.1.

#### 3.9.2.1 research/malware-domain

Returns a JSON response containing static details about matching samples as well as antivirus vendor and signature details.
Due to the recursive nature of this method, the number of responses provided will be limited by maximum response time.

**Table 3.23 - API to request information on a domain**

| research/malware-domain | |
|---|---|
| Method | POST |
| Parameters | Required fields:<br>• apikey : string : your api key<br>• domain : string or list : domain to query<br>Optional fields:<br>• limit : integer : maximum number of samples to return per domain (1..1000); default 100 |
| Response | JSON object containing:<br>• domain : domain that was queried |

| | |
|---|---|
| | • family : Shadowserver family classification<br>• tag : whether detected as a DGA by the TNO Classifier (DGA Detective)<br>• samples : list of samples that queried for the domain, along with the sample's MD5 hashes, SHA1 hashes, SHA256 hashes, SHA512 hashes, AV vendor name and detections |
| Example Request | `{ "domain": "nhnrqmpwnr.us" }` |
| Example Response | `{ "nhnrqmpwnr.us": { "family": "mydoom", "tag": "tno-dga-tagged", "samples": [ { "md5": "fda0344324c77ea2c2c954489de3e0e4", "sha1": "6ded10575668deee9b79a2777346b7c7597db7d0", "sha256": "78e9f9133488c96c2933fea15e04a4c981c957029dfa7e5f9bce214a843624a2", "anti-virus": [ { "vendor": "Avast", "signature": "Win32:Mydoom-BJ" } ] }, { "md5": "e62bf32812837d88e465d8ec69f5999d", "sha1": "820176d8d38369b4b903ebeca08287179c7c6de8", "sha256": "5d1a2b219abbcd0fbff629979c7f15d4688bb7a7211e0e1bfdbe6ce0d650831f", "anti-virus": [ { "vendor": "Avast", "signature": "Win32:Mydoom-BJ" } ] }, { "md5": "6d4a323ef1a1519d816d445ae478cb29", "sha1": "fb4b3c46e2c829cb326f7d43b09a2f1a28c92f4d", "sha256": "782ab8682dfdc432064523e36d2f731ec92f7758f5ee677926b27cd2b9822fce", "anti-virus": [ { "vendor": "Avast", "signature": "Win32:Mydoom-BJ" } ] } ] } }` |

### 3.9.2.2 research/malware-info

Returns a JSON response containing static details about the requested sample as well as DNS and antivirus vendor and signature details. Domains that are detected as potential DGAs by the TNO classifier will have an additional tag present.

**Table 3.24 – API to request information on a (hash of) malware sample**

| research/malware-info | |
|---|---|
| Method | POST |
| Parameters | • apikey : string : your api key<br>• sample : string : MD5, SHA1, SHA256, and SHA512 hash |
| Response | JSON object containing:<br><br>• magic : file magic number<br>• filesize : file size in bytes<br>• tlsh : tlsh hash value<br>• adobe_malware_classifier : Adoble malware classifier tool verdict: “clean” “malicious” or “unknown”<br>• timestamp : timestamp of when the sample was processed<br>• first_seen : timestamp when sample was first seen<br>• last_seen : timestamp when sample was last seen<br>• MD5 : MD5 hash of sample<br>• SHA1 : SHA1 hash of sample<br>• SHA256 : SHA256 hash of sample<br>• SHA512 : SHA512 hash of sample<br>• type : type of file<br>• domain_name : list of domain names (if any), with TNO classifier (DGA Detective) tags when available<br>• anti_virus : list of AV vendors and detections (if any) |
| Example Request | `{ "sample": "6ded10575668deee9b79a2777346b7c7597db7d0" }` |
| Example Response | `[ { "magic" : "PE32 executable (GUI) Intel 80386 (stripped to external PDB), for MS Windows", "sha512" : "d035239697018cbda448d15c3598cfe2fde1f18f39974e66db9b8c907896067d56bcf7cf5d6fc0a57fe89063a272bc3341fdf29ebf2ac70a6c44cd1b22d336ef", "sha1" : "6ded10575668deee9b79a2777346b7c7597db7d0", "tlsh" : "a9c34b1af50255f7ef2908b002c7f2be4a20ae24d4244e86ec73de5766d353", "adobe_malware_classifier" : "unknown", "filesize" : "121376", "timestamp" : "2021-04-26 02:47:09", "sha256" : "78e9f9133488c96c2933fea15e04a4c981c957029dfa7e5f9bce214a843624a2", "first_seen" : "2021-04-26 02:47:09", "last_seen" : "2021-04-26 02:47:09", "type" : "exe", "md5" : "fda0344324c77ea2c2c954489de3e0e4" "domain_name" : [ { "domain" : "mail.hope-mail.com" }, { "domain" : "mx.cam.ac.uk" }, { "domain" : "freedownloadmanager.org"` |

```
        },
        {
           "domain" : "p01mgw03.macromedia.com",
           "tag" : [
              "tno-dga-tagged"
           ]
        },
        {
           "domain" : "apple.com"
        },
        {
           "domain" : "alt3.gmail-smtp-in.l.google.com"
        },
        {
           "domain" : "mx156.hostedmxserver.com"
        },
        {
           "domain" : "pb-mx11.pobox.com"
        },
        {
           "domain" : "firefox.mozilla.org"
        },
        {
           "domain" : "torntv.com"
        },
        {
           "domain" : "ask.com"
        },
        {
           "domain" : "adsremoval.net"
        },
        {
           "domain" : "alt3.aspmx.l.google.com"
        },
        {
           "domain" : "mxb-00377f01.gslb.pphosted.com",
           "tag" : [
              "tno-dga-tagged"
           ]
        },
        {
           "domain" : "nnepqerhnh.us",
           "tag" : [
              "tno-dga-tagged"
           ],
           "family" : "mydoom"
        },
        {
           "domain" : "anchorfree.com"
        },
        {
           "domain" : "ylgga.com"
        },
        {
           "domain" : "mail1.internetdownloadmanager.com"
        },
        {
           "domain" : "mybrowserbar.com"
        },
        {
           "domain" : "bulletmedia.com"
        },
```

```
            {
               "domain" : "northcoast.com"
            },
            {
               "domain" : "flash.com"
            },
            {
               "domain" : "wnhaweaana.in",
               "tag" : [
                  "tno-dga-tagged"
               ],
               "family" : "mydoom"
            },
            {
               "domain" : "aspmx.l.google.com"
            },
            {
               "domain" : "pwprehsnmh.in",
               "tag" : [
                  "tno-dga-tagged"
               ],
               "family" : "mydoom"
            }
         ],
         "anti_virus" : [
            {
               "signature" : "Win32:Mydoom-BJ",
               "vendor" : "Avast"
            },
            {
               "signature" : "Win32/Heur.dropper",
               "vendor" : "AVG"
            },
            {
               "signature" : "Trojan.GenericKDZ.66635",
               "vendor" : "BitDefender"
            },
            {
               "signature" : "TR/Proxy.Gen",
               "vendor" : "Avira"
            },
            {
               "signature" : "Trojan ( 004d7c651 )",
               "vendor" : "K7GW"
            },
            {
               "signature" : "W32/S-e4365596!Eldorado",
               "vendor" : "Authentium"
            },
            {
               "signature" : "Trojan ( 004d7c651 )",
               "vendor" : "K7"
            },
            {
               "signature" : "Trojan.GenericKDZ.66635",
               "vendor" : "GData"
            },
            {
               "signature" : "Trojan.Win32.Mydoom",
               "vendor" : "Ikarus"
            },
            {
```

```
            "signature" : "Trojan:Win32/Mydoom",
            "vendor" : "Microsoft"
         },
         {
            "signature" : "W32/Mytob.gen@MM.i",
            "vendor" : "McAfee"
         },
         {
            "signature" : "Win32/Agent.NHB",
            "vendor" : "Eset"
         },
         {
            "signature" : "Trojan.Win32.Generic!BT",
            "vendor" : "Sunbelt"
         },
         {
            "signature" : "Trojan.TR/Proxy.Gen",
            "vendor" : "FSecure"
         }
      ]
   }
]
```


### 3.9.2.3 research/tno-export

List all domains classified as DGAs by the TNO classifier (DGA Detective) from Sandbox events in the past day. Optionally export as snort rules.

**Table 3.25 – API to request TNO classifier (DGA Detective) results from past day**

| **research/tno-export** | |
|---|---|
| Method | POST |
| Parameters | Required Field:<br>• apikey : string : your api key<br>Optional Fields:<br>• tld : string : top level domain filter<br>• family : string : family filter ; when supported by the TNO classifier<br>• limit : integer : maximum number of responses to return<br>• skip : string : exclude results that match known DGA families, like “rpz-dga”<br>• format : string : set to "snort" for snort rules (default format is JSON) |
| Response | JSON lines containing<br>• tld – tld of domain tagged by the TNO classifier (DGA Detective)<br>• domain - domain tagged by the TNO classifier (DGA Detective).<br>Or if snort format chosen<br>• snort rule for each domain listed, with domain name and date when rule was generated |
| Example Request (1) | - |
| Example Response (1) | `{"tld":"com","domain":"www.brt0wgrpd1.com"}`<br>`{"tld":"pw","domain":"cxxeddhauqls.pw"}`<br>`{"tld":"com","domain":"www.vaftowqmjf.com"}` |
| Example Request (2) | `{"tld":"info","skip":"rpz-dga","format":"snort","limit":3}` |

| | |
|---|---|
| Example Response (2) | alert udp $HOME_NET any -> any 53 (msg:"2021-08-20 TNO Classifier DGA tagged domain: wzamztlunpgljmsf.info"; content:"\|01 00 00 01 00 00 00 00 00 00\|"; depth:10; offset:2; content:"\|10\|wzamztlunpgljmsf\|04\|info\|00\|"; fast_pattern; nocase; distance:0; content:!"\|0010\|wzamztlunpgljmsf\|04\|info\|00\|"; nocase; classtype:trojan-activity; sid:3000000; rev:1;)<br>alert udp $HOME_NET any -> any 53 (msg:"2021-08-20 TNO Classifier DGA tagged domain: pgjzyqbsmlepki.info"; content:"\|01 00 00 01 00 00 00 00 00 00\|"; depth:10; offset:2; content:"\|0e\|pgjzyqbsmlepki\|04\|info\|00\|"; fast_pattern; nocase; distance:0; content:!"\|00 0e\|pgjzyqbsmlepki\|04\|info\|00\|"; nocase; classtype:trojan-activity; sid:3000001; rev:1;)<br>alert udp $HOME_NET any -> any 53 (msg:"2021-08-20 TNO Classifier DGA tagged domain: npznuxfxiksqfowq.info"; content:"\|01 00 00 01 00 00 00 00 00 00\|"; depth:10; offset:2; content:"\|10\|npznuxfxiksqfowq\|04\|info\|00\|"; fast_pattern; nocase; distance:0; content:!"\|00 10\|npznuxfxiksqfowq\|04\|info\|00\|"; nocase; classtype:trojan-activity; sid:3000002; rev:1;) |

### 3.9.2.4 research/rpz-export

Returns the list of active RPZ (Response Policy Zone) DGA domains valid for the next five days.

**Table 3.26 – API to request active RPZ DGA domains**

| **research/rpz-export** | |
|---|---|
| Method | POST |
| Parameters | Required Field:<br>• apikey : string : your api key<br>Optional Fields:<br>• date : string : date or date range ; must begin within the past 10 days<br>• tld : string : top level domain filter<br>• family : string : family filter<br>• limit : integer : maximum number of responses to return |
| Response | JSON lines containing entries such as:<br>• tag information, such as rpz-dga<br>• timestamp: timestamp of validity period of DGA (UTC)<br>• family: DGA family classification<br>• domain: DGA domain<br>• rule: DGA domain seed<br>• source: source of DGA information<br>• tld: tld of DGA domain |
| Example Request | - |
| Example Response | {"tag":["rpz-dga"],"timestamp":"2021-04-20 00:00:00","family":"bazarloader","tld":"bazar","timeend":"2021-04-20 00:00:00","domain":"cabaudwy.bazar","rule":"bazarloader_dga_3","source":"Fraunhofer FKIE DGArchive"}<br>{"source":"Fraunhofer FKIE DGArchive","rule":"bazarloader_dga_3","timeend":"2021-04-20 00:00:00","domain":"caeqtoek.bazar","tld":"bazar","family":"bazarloader","timestamp":"2021-04-20 00:00:00","tag":["rpz-dga"]}<br>{"timestamp":"2021-04-20 00:00:00","family":"bazarloader","tag":["rpz-dga"],"timeend":"2021-04-20 00:00:00","domain":"caynibwy.bazar","rule":"bazarloader_dga_3","source":"Fraunhofer FKIE DGArchive","tld":"bazar"} |

| | |
|---|---|
| | {"tag":["rpz-dga"],"timestamp":"2021-04-20 00:00:00","family":"bazarloader","tld":"bazar","rule":"bazarloader_dga_3","source":"Fraunhofer FKIE DGArchive","timeend":"2021-04-20 00:00:00","domain":"cakitoid.bazar"} |

### 3.9.2.5 research/site-ranking-export

Provides domain name ranking details compiled from a number of public sources.

**Table 3.27 – API to request latest domain ranking**

| research/site-ranking-export | |
|---|---|
| Method | POST |
| Parameters | Required Field:<br>• apikey : string : your api key<br>Optional Fields:<br>• tld : string : top level domain filter<br>• source : string : source filter<br>• position : integer : minimum position filter<br>• limit : integer : maximum number of responses to return |
| Response | JSON lines containing<br>• domain : domain name on a domain ranking list<br>• source : source of the ranking list<br>• position : position on the ranking list |
| Example Request | - |
| Example Response | { "domain": "window2baku.com", "source": "domcop", "position": 209119 }<br>{ "domain": "lexique-du-net.com", "source": "domcop", "position": 209182 }<br>{ "domain": "dashboards.trefis.com", "source": "domcop", "position": 209207 } |

### 3.9.2.6 research/sandbox-dns-anomaly

Provides sandbox DNS alert details for one or more types query types: crypto (DoH, DoT usage), registrar and TLD trend analysis.

**Table 3.28 – API to request sandbox DNS alert details**

| research/sandbox-dns-anomaly | |
|---|---|
| Method | POST |
| Parameters | Required Field:<br>• apikey : string : your api key<br>• type : string or list : [ "crypto", "registrar", "tld" ]<br>Optional Fields:<br>• limit : integer : maximum number of responses to return<br>• date : string : date or date range |
| Response | JSON object<br>Values returned: JSON objects (tld, registrar, crypto) |

| | |
|---|---|
| | For tld objects:<br>• timestamp<br>• tld : TLD for tld objects<br>• resolv : whether a domain resolves or not<br>• type : type of TLD, for example, ccTLD<br>• queries : queries seen<br>• percent : percentage increase versus Avg7<br>• avg7: rolling 7 days average queries<br>• avg14 : rolling 14 days average queries<br>• avg30 : rolling 30 days average queries<br>• avg90 : rolling 90 days average queries<br>• avg120 : rolling 120 days average queries<br>• avg180 : rolling 180 days average queries<br>• registrar - name of DNS registrar (for registrar queries)<br>• type – doh or dot (for crypto queries)<br>• address – server address (for crypto queries) |
| Example Request | `{ "type": [ "tld", "registrar" ] }` |
| Example Response | `{ "tld": [ { "timestamp": "2021-05-02", "tld": "ua", "resolv": "no", "type": "ccTLD", "queries": 6, "avg7", 2, "percent": 272, "avg14": 1, "avg30": 1, "avg90": 2, "avg120": 2, "avg180": 2, "avg365": 2 } ], "registrar": [ { "timestamp": "2021-05-02", "registrar": "Akamai Technologies, Inc.", "queries": 547, "avg7", 233, "percent": 234, "avg14": 202, "avg30": 287, "avg90": 1472, "avg120": 1472, "avg180": 1472, "avg365": 1472 } ] }` |

## 3.10 Automated Reconfiguration

### 3.10.1 Functional specification

The Automated Reconfiguration (AR) component provides the capability to interface with IT support or IT infrastructures directly to reconfigure security controls based on the selected response actions to mitigate an emerging threat or detected attack. The component will provide a generic and extendible framework to interface with multiple types of system, using open standards such as OASIS OpenC2 and OASIS CACAO.

The functions provided by the (Automated) Reconfiguration Component are:

- Plan and prepare a CoA into individual reconfiguration actions (generate a *CoA execution playbook*).
- Request authorisation for reconfiguration action from human operator.
- Request manual reconfiguration by human operator via e.g. IT support ticketing system.

- Execute reconfiguration actions by interfacing directly with network element (e.g. firewalls, routers), endpoints and/or management systems (e.g. Active Directory, SDN controller, Docker, Kubernetes).
- Translate the CoA language used in SOCCRATES to CoA description language and/or playbook description language for execution of CoA by third party Security Orchestration, Automation and Response (SOAR) product.
- Request CoA execution by third party SOAR product.

We expect that, for the coming years, most organisation will require a human in the loop to authorize reconfiguration. As such SOCCRATES Platform has focussed on opening an IT support ticket, and explore the preparation of fully automatic reconfiguration using OpenC2 commands and CACAO playbooks.

### 3.10.2 API specification

The Automated Reconfiguration (AR) component consists of a workflow running on Activiti and a set of Cortex responders. Unlike the rest of the components that are part of the SOCCRATES workflow, the AR triggers actions through external tools, including opening tickets on an IT support ticketing system for dealing with the response, sending notification emails or triggering OpenC2 commands. This is depicted in Figure 3.11.

Depicted are three types of Cortex responders:

- **Email Cortex responder** – this can be used to request authorisation, requesting manual reconfiguration, or opening a ticket via email interface.
- **IT support ticket Cortex responder** – this can be used to directly open a ticket on an IT support ticketing system (for requesting authorisation and/or manual reconfiguration).
- **OpenC2 proxy Cortex responder** – this can be used to translate an OpenC2 command to a specific configuration command for a security function.

Other types of Cortex responders are possible, such as a Cortex responder that provides (or translates) the CACAO playbook, prepared by the Containment CoA Generator Cortex analyser of the Response Planner, to a third-party Security Orchestration, Automation and Response (SOAR) product.

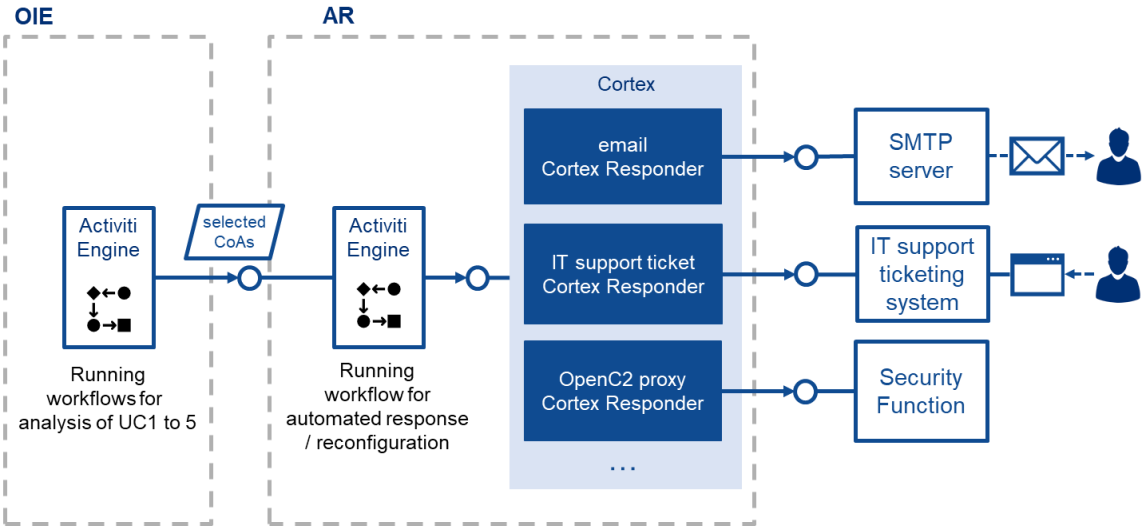


**Figure 3.11 – Integration architecture for AR**

The current and public Cortex community provides with several responders to interact with different platforms, including free[13] and subscription-based ones[14]. Among the free responders, already available directly from the Cortex Github account, is a Redmine responder. It allows to automatically open issues, allowing also to assign responsible person of dealing with the issue or closing the issue when the task is completed.

Another responder available in the default Cortex installation provides the capability to send emails just by configuring the SMTP server and credentials. This is also useful for the AR when notifying about response actions to certain people. The configuration for Mailer Cortex responder is given below:

```
#Configuration file for the Mailer Cortex Responder
{
  "name": "Mailer",
  "version": "1.0",
  "author": "CERT-BDF",
  "url": "https://github.com/TheHive-Project/Cortex-Analyzers",
  "license": "AGPL-V3",
  "description": "Send an email with information from a TheHive case or alert",
  "dataTypeList": ["thehive:case", "thehive:alert"],
  "command": "Mailer/mailer.py",
  "baseConfig": "Mailer",
  "configurationItems": [
    {
      "name": "from",
      "description": "email address from which the mail is send",
      "type": "string",
      "multi": false,
      "required": true
    },
    {
      "name": "smtp_host",
      "description": "SMTP server used to send mail",
      "type": "string",
      "multi": false,
      "required": true,
      "defaultValue": "localhost"
    },
    {
      "name": "smtp_port",
      "description": "SMTP server port (default 25, SSL 465, STARTTLS 587)",
      "type": "number",
      "multi": false,
      "required": true,
      "defaultValue": "25"
    },
    {
      "name": "username",
      "description": "Username to use in case of SSL/STARTTLS",
      "type": "string",
      "multi": false,
      "required": true
    },
    {
      "name": "password",
      "description": "Password to use in case of SSL/STARTTLS",
```

[13] https://github.com/TheHive-Project/CortexDocs/blob/master/analyzer_requirements.md#free-responders

[14] https://github.com/TheHive-Project/CortexDocs/blob/master/analyzer_requirements.md#subscription-and-license-based-responders

```
        "type": "string",
        "multi": false,
        "required": true
      }
    ]
}
```

There currently is a License-based Cortex responder available for the Cisco AMP for Endpoints[15] that can be for instance used for containment of compromised hosts via:

- AMPforEndpoints_IsolationStart: Start Host Isolation.
- AMPforEndpoints_IsolationStop: Stop Host Isolation.

Regardless the Cortex responders already available, the flexibility of the Cortex framework allows to develop responders very easily and integrate additional Cortex responders by using APIs already available at certain platforms. Such is the case of JIRA tickets, which can be opened and managed through a REST-API[16].

To demonstrate automated reconfiguration at the demonstration pilot site, the SOCCRATES project intends to develop a proof-of-concept OpenC2 proxy Cortex responder. At the moment of writing this deliverable, the SOCCRATES project is determining which type of Containment CoA can be automatically executed given the available security control in the demo pilot. For example, containment of a compromised host can be done by using Endpoint Detection and Response (EDR).

## 3.11 Web Front End

Unlike other components of the SOCCRATES architecture, the Web Front End (WFE) does not provide with a specific security analysis or response function by itself. Instead, it feeds information obtained from the execution of the workflows at the OIE and from a database that contain information from previous workflow executions, to show relevant information to SOC analyst either for supporting decisions that are part of the OIE or to get additional insights regarding some of the events processed by the OIE workflows (i.e., incidents, new vulnerabilities, etc). To this end, the WFE also links GUIs from tools that are part of the SOCCRATES workflows, such as the ADG. Figure 3.12 presents a conceptual overview of the Web Front End.

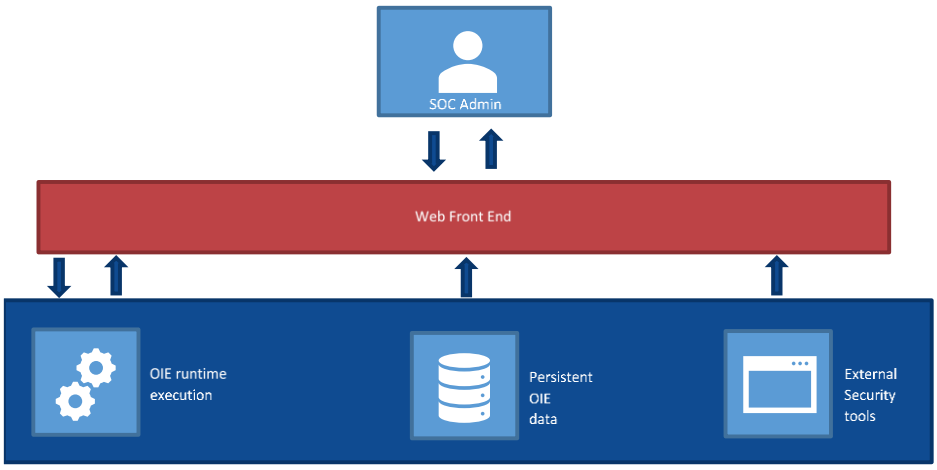


**Figure 3.12 – Web Front End inputs sources and output consumers**

[15] AMP for Endpoints was the former name. Current name for the product is Cisco Secure Endpoint.
[16] https://developer.atlassian.com/server/jira/platform/rest-apis/

### 3.11.1 Functional specification

As described in D6.1, there are two main functions that the WFE is responsible for:

- Follow up the current status of a workflow execution. For a given triggering event (i.e., incident, vulnerability found, etc) the WFE provides with a real time snapshot of the workflow managing such event. It provides with a graphical representation of the workflow being executed, the stage where the execution is, and information detailing the results of the previous stages.
- Display information from external security tools, by integrating their GUIs in the WFE, allowing SOC administrators to have additional information for further analysis of the security incidents, the status of the infrastructure, etc.

As depicted in Figure 3.12, the WFE obtains information from different sources:

- From the OIE Runtime execution. When a workflow is executed in the Activiti engine a set of data is available in real time, constantly updated as long as the execution progresses. This includes data such as environment variables that are created at execution time and mapped to the result obtained from the execution of the tasks of the workflow. These variables are dynamically created at execution time. The WFE reads those variables to display relevant information to the SOC analysts (i.e., the status of execution of one of the tasks of the workflow). Additional information that can be requested at the OIE Runtime execution is the diagram of the workflow, the stage of execution, users involved, version of the workflow, tasks involved, etc. All this information can be obtained from the OIE by using its REST API, as described in Section 3.2.2.
- Persistent data from the OIE. When a workflow execution ends and before the instance of the process execution is removed from memory, some resulting data is stored in a database to keep track of execution results, data exchanged with external security tools and any other information that could be useful on a later stage (for instance, for forensics activities). This data is also available for the WFE, which can be used to provide information about past events to the SOC analyst.
- Link to external security tools. From the WFE the SOC analyst can be forwarded to GUIs from external security tools that are part of the workflows at the OIE. This is useful for SOC analysts to extend the information displayed when taking decisions. To this end, the WFE integrates the GUIs from external tools into the same front end, managing sessions, roles and authentication to provide SOC analysts with a seamless user experience.

Additionally, the SOC analyst can interact through the WFE with the OIE runtime execution to make decisions based on the information displayed at the WFE. As described in D6.1, the OIE workflows can include human in the loop tasks which allow SOC analysts to decide on several aspects of the execution, such as in UC1 when determining whether an incident requires a containment strategy after an initial analysis. The WFE can interact with the OIE runtime execution through the Activiti REST API, as described in Section 3.2.2.

# 4 Use cases

## 4.1 Introduction

In this chapter we describe how the SOCCRATES platform is applied in each of the use cases. It is important to notice that Use Cases 2 to 5 describe a pre-attack phase, with the aim of improving the cyber resilience of the organization. The organization is not under active attack yet, and there is no active adversary. Use Cases 2 to 5 aim to increase the security posture of the organization and prevent successful attacks from happening.

Use Case 1 on the other hand, describes an active attack that requires incident response activities. The difference in stages between Use Case 1 and Use Cases 2 to 5 was described in section 2.1 and depicted in Figure 2.2.

The Use Cases and swim lanes, initially described in deliverable D2.1 [1], have been updated to reflect the lessons learned during the SOCCRATES project. For instance, the threat hunting activity in Use Case 1 was deemed too difficult to do in a fully automated way with the current state-of-the-art. Therefore, it was integrated as a manual parallel task that the threat hunting team could perform, supported by the information collected and presented by the SOCCRATES platform.

The steps how SOCCRATES platform is applied in Use Cases 2 to 5 can be described in a single combined swim lane diagram. Use case 1, however, is much more complicated and consists of five swim lane diagrams, see Figure 4.1. In the next sections all the steps in the swim lane diagrams are described.

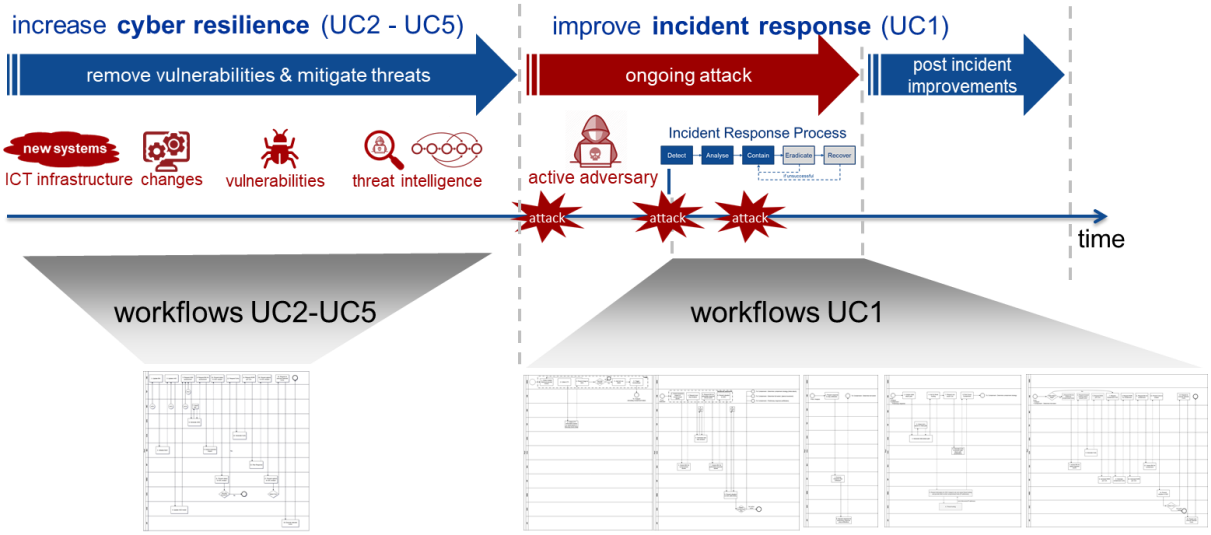


**Figure 4.1 – Difference in stages between Use Case 1 and Use Cases 2 to 5, and the swim lane diagrams for the use cases.**

All interactions in the swim lane diagrams originating from the OIE to a component are made through Cortex, by calling Cortex analysers or Cortex responders for the component.

## 4.2 Use Case 1

The general outline of Use Case 1 has remained unchanged compared to D2.1 [1]. The workflow of Use Case 1 starts with the detection of an attack by the SIEM. If the detection was triggered by a

security event from the AI-based Attack Detection (AAD) component, the stage of the attack (such as initial access, lateral movement, and exfiltration) may be included in the security event and included in the triggering of the workflow by SIEM (see section 3.1.1). After the analysis of the current situation, the containment of the incident starts, using the stage of the attack provided by the AAD to start different sub-workflows. In case of the detection of an exfiltration, a preliminary response workflow will be started. After this preliminary response, or when the AAD discovered a lateral movement, the SOCCRATES platform will determine the full extent of the attack in a sub-workflow. Finally, for all attack stages, the final containment sub-workflow will be started.

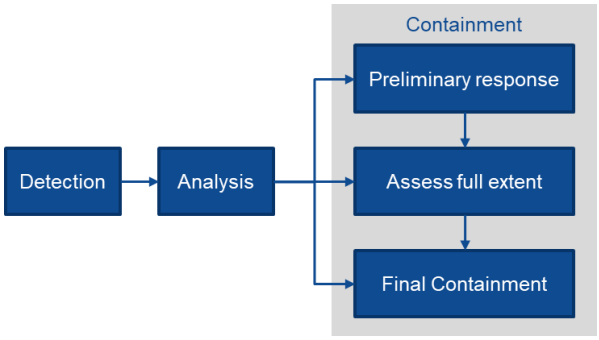


**Figure 4.2 – Sub-workflows for Use Case 1.**

### 4.2.1 Detection

In Figure 4.3 the swim lane diagram is depicted for the detection stage.

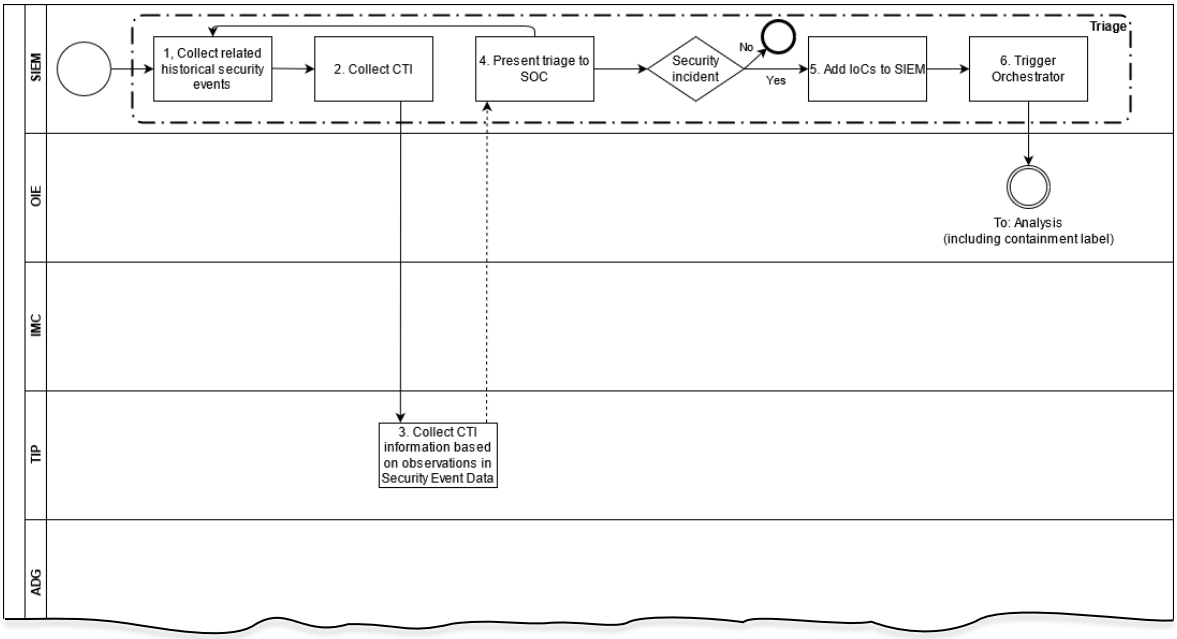


**Figure 4.3 – Swim lane diagram for detection phase of Use Case 1.**

Although the SIEM solution sits outside the scope of the SOCCRATES project, different components of the SOCCRATES platform need to be able to interact with it. At both pilot sites (i.e. Vattenfall and mnemonic) there are existing SIEM and TIP solution that have already been integrated to some extent. It was therefore decided to rely on existing processes and integration facilitating the collection of CTI to support triage and trigger the SOCCRATES platform after the SOC analyst classified the SIEM

alert as a security incident. Note that these steps can also be supported by the SOCCRATES platform (i.e. workflow in the OIE).
Use Case 1 is started by a security event that triggers an alert in the SIEM. This can be information sent via syslog, third party client software, or any other SIEM input event. Within the SOCCRATES platform, another option is that it is a security event from the AI based Attack Detection (AAD). When triggered based on a security event from the AAD, the trigger comes in with an additional label for the event type (i.e. initial access, lateral movement, or exfiltration) that was determined by the AAD.

The steps of the workflow depicted in the swim lane diagram are:

1. One of the primary functions of a SIEM is to correlate security events. The SIEMs at both pilot sites collect related historical security events.
2. The SIEM will then try to match the collected security events with CTI information from a TIP to enrich the security event.
3. The TIP receives the observed indicators in the security event and matches that with available threat intelligence, such as additional Indicators of Compromise (IoCs). The TIP returns the threat intelligence to enrich the security information for the SOC analyst.
4. The threat intelligence (including additional IoCs) is collected by the SIEM and presented to the SOC analyst for triage. The SOC analyst can decide to do another cycle of querying the SIEM or the TIP to collect more information before making the decision. Based on the enriched security incident data, the SOC analyst may decide that it is a security incident. If not, then the workflow stops here. Else it will continue as follows.
5. The additional IoCs collected from the TIP are added to the SIEM, to make sure that future events related to the security incident will be filtered and added to the incident.
6. As a final stage of the triage phase, the OIE is triggered. The security incident is forwarded to the orchestrator, together with the attack stage (or event type) determined by the AAD (initial access, lateral movement, or exfiltration).

### 4.2.2 Analysis

In the figure below the swim lane diagram is depicted for the analysis stage, creating situational awareness. During the analysis phase, the SIEM sends all security incident data together with the event type to the REST API of the OIE, thus starting the workflow in the Activiti orchestrator.

1. The OIE will start with assessing the impact for the impacted IP address(es) that were discovered by the SIEM using the "*bia_analysis*" function of the Cortex analyser.
2. On request from the OIE, the BIA will assess the business impact for all compromised assets, and return the results from the "bia_analysis" function to the OIE.
3. From a defence perspective, it is interesting to know how the attacker can reach critical assets from the compromised hosts. First, this can support the threat hunting activities during the incident response, by returning the most probable attack paths. The SOC analyst could focus on the assets on this path, by looking for indicators of compromised for these assets. Second, it could help the SOC team to prevent critical assets from being compromised during the attack by applying additional countermeasures on the most probable attack path. Therefore, the OIE calls the ADG for a simulation.

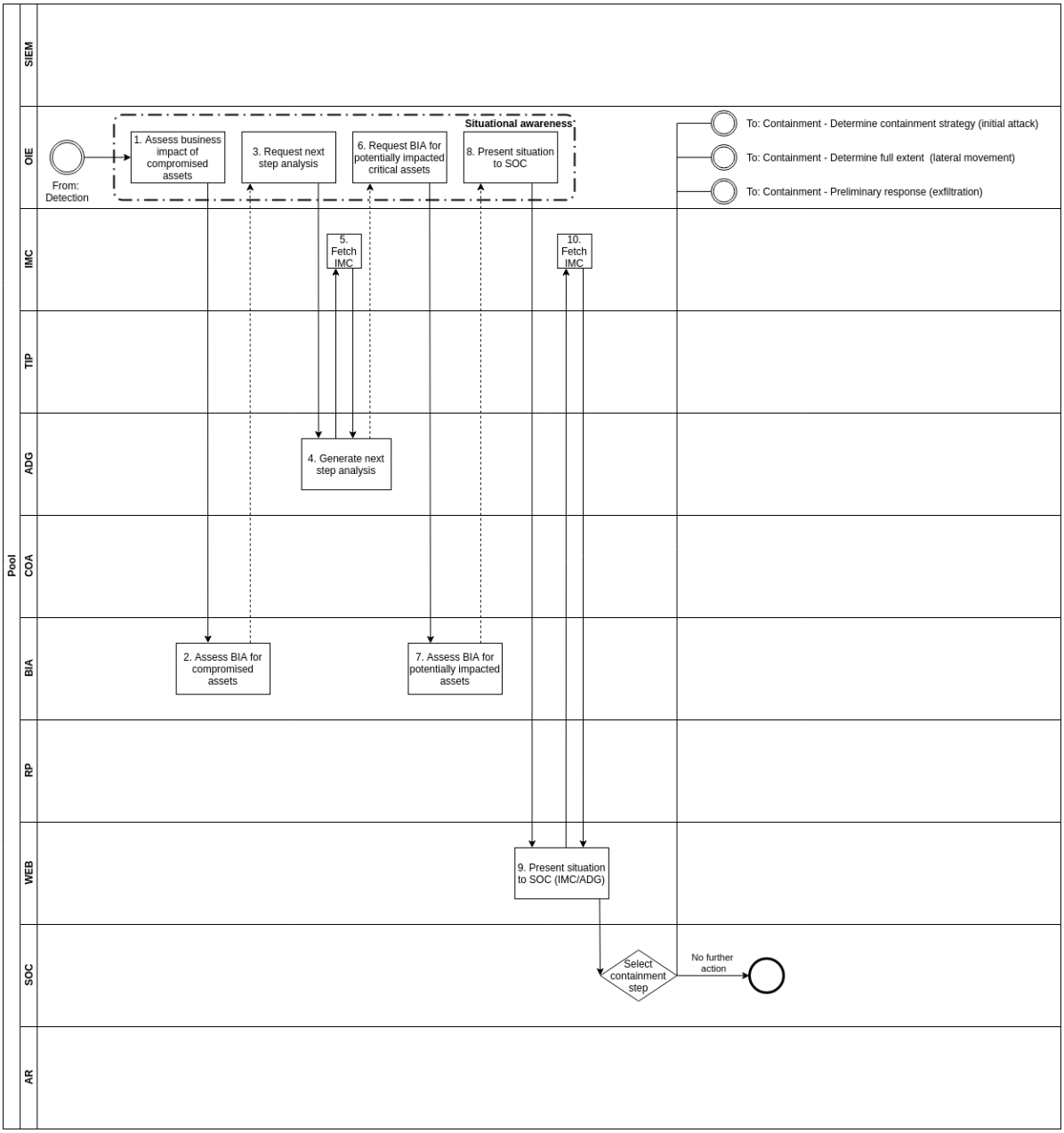


**Figure 4.4 – Swim lane diagram for analysis phase of Use Case 1.**

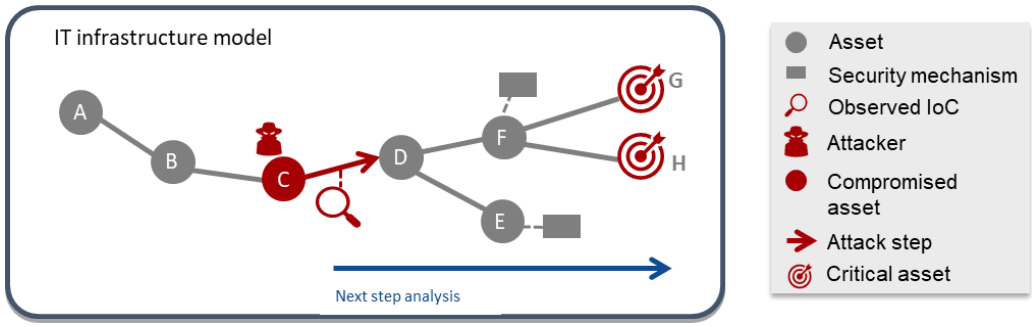


**Figure 4.5 – Conceptual visualisation of next step analysis with the ADG.**

4. In case there is evidence that asset C in Figure 4.5 is compromised, a next step analysis is performed to check how the attacker would be able to move from asset C to the critical assets

(G and H) in the infrastructure. The parameters for this ADG analysis are given in Table 4.1. The attacker is placed on asset C, with the critical assets G and H as highValueAssets (i.e. retrieved from the IMC). If the APT is known from the results of the TIP, this information could be added to the ADG *simulation* call (section 3.4.3). If specific CVEs were found to be used during the attack, these could be added to the analysis as well.

**Table 4.1 - Parameters for ADG simulation for next step analysis**

| Simulation Parameter | Next step analysis |
|---|---|
| date | Date of successful attack (retrieved from SIEM security incident information) |
| threatActor | Threat actor from TIP, or default if unknown |
| entryPoints | Asset C |
| highValueAssets | - |
| vulnerabilities | Derived from SIEM security information, or default (all) if unknown |

5. The ADG will use the infrastructure model available in the IMC. Since this infrastructure model may become very large over time, the ADG will directly connect to the API of the IMC with the configured the *imc_url*, *imc_username* and *imc_password*, instead of making the request through the OIE.
6. The OIE will store the results from the ADG and call the BIA to analyse the impact for the potentially impacted critical assets based on the ADG results. The purpose of doing this is for the SOC analyst to get situational awareness of the incident. What assets in the infrastructure are potentially hit, what is the most probable path, how much effort (TTC) would it take for the attacker to get to the critical assets, and what would be the business impact.
7. The same "*bia_analysis*" function of the Cortex analyser is called as in step 2, but this time with the result of the ADG, containing *assetID* (a unique reference to an asset in the IMC), supported by TTC and probability of the confidentiality, integrity and availability (section 3.6.2). The BIA calculation will be performed for the entire company. The attack time can be set to the time of the security event data received by the SIEM.
8. The BIA component gives its response back to the OIE. The OIE will present the situation to the SOC analyst via the web interface.
9. The web interface will use the IMC containing the compromised and potentially impacted assets, to show the current situation of the infrastructure. It will also contain a link to see the full report of the ADG based on the "*report_url*". The compromised and potentially compromised hosts are presented to a SOC analyst together with situational information.
10. The situation of the compromised assets and potentially impacted assets is presented to the SOC analyst with the graphical interface of the IMC, the *report_url* provided by the ADG, and results of the BIA.
11. The SOC analyst is asked to select the containment step (initial attack, lateral movement, or exfiltration) or to stop the workflow by selecting "*no further action*". This selection is forwarded to the OIE with the containment step to select the next workflow.

### 4.2.3 Containment – Preliminary response

In the Figure 4.6, the swim lane diagram is depicted for the Containment stage – preliminary response. When exfiltration of company data is detected and confirmed by the SOC analyst, a quick

preliminary response is necessary to stop the exfiltration as soon as possible to prevent further damage. This may provide information to the attacker that some of his/her actions have been detected.

1. In case the OIE is triggered to request a CoA to stop exfiltration via the “generateContainmentCoas” Cortex analyser from the RP.
2. The RP Cortex will create the right OpenC2 commands based on the provided IP address(es) and exfiltration containment CoA and wrap this in a CACAO playbook. This is returned to the OIE
3. The OIE will send the CACAO playbook for execution to the AR
4. The AR extracts the CACAO playbook and executes the OpenC2 command in the playbook to handle the exfiltration. It will return the response of the OpenC2 consumer to the OIE, indicating whether it was successfully executed.

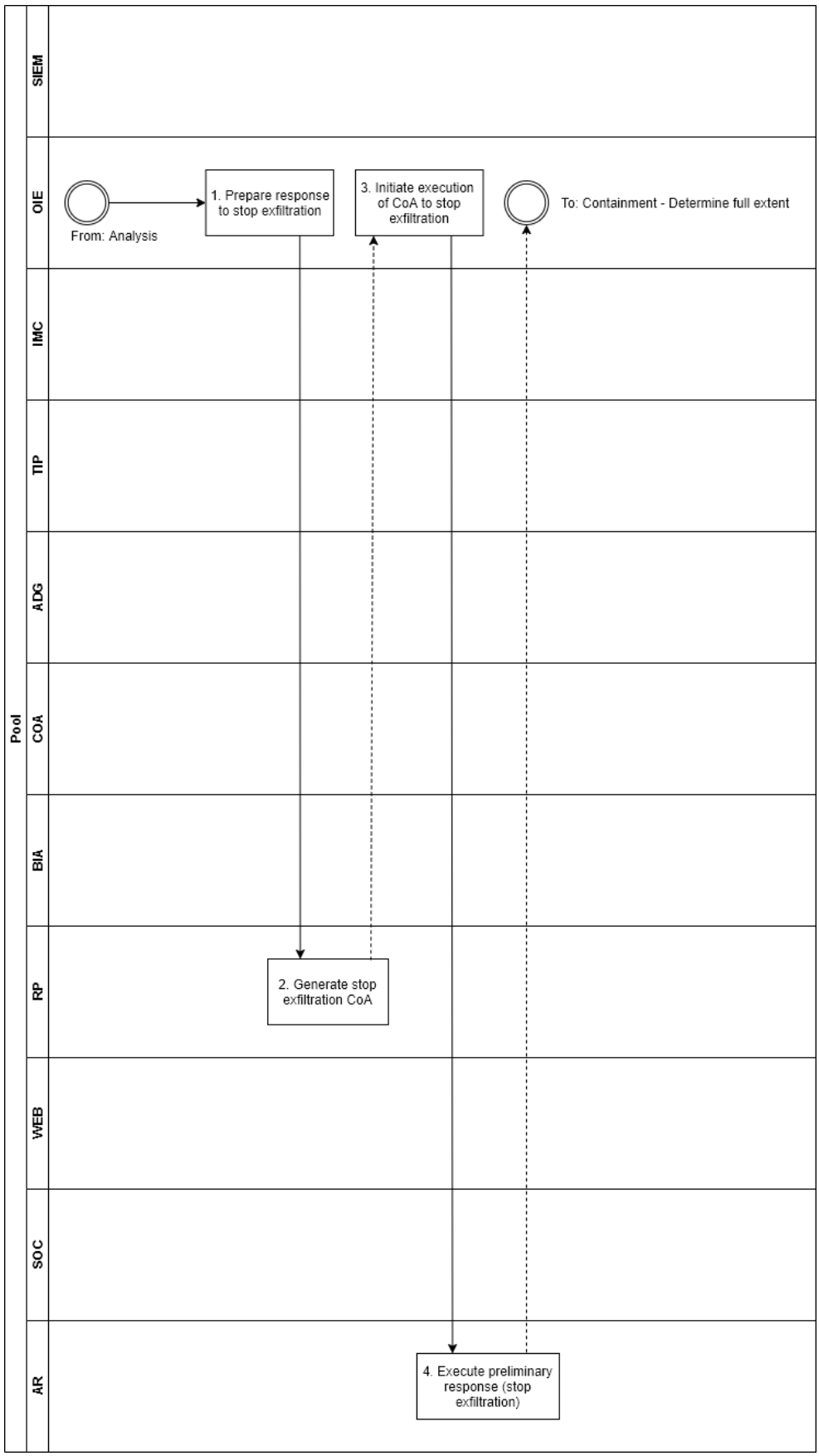


**Figure 4.6 – Swim lane diagram for preliminary response in containment phase of Use Case 1.**

### 4.2.4 Containment – Assess full extent

In Figure 4.7 the swim lane diagram is depicted for the assessment of the full extent of the security incident as part of the containment phase of Use Case 1.

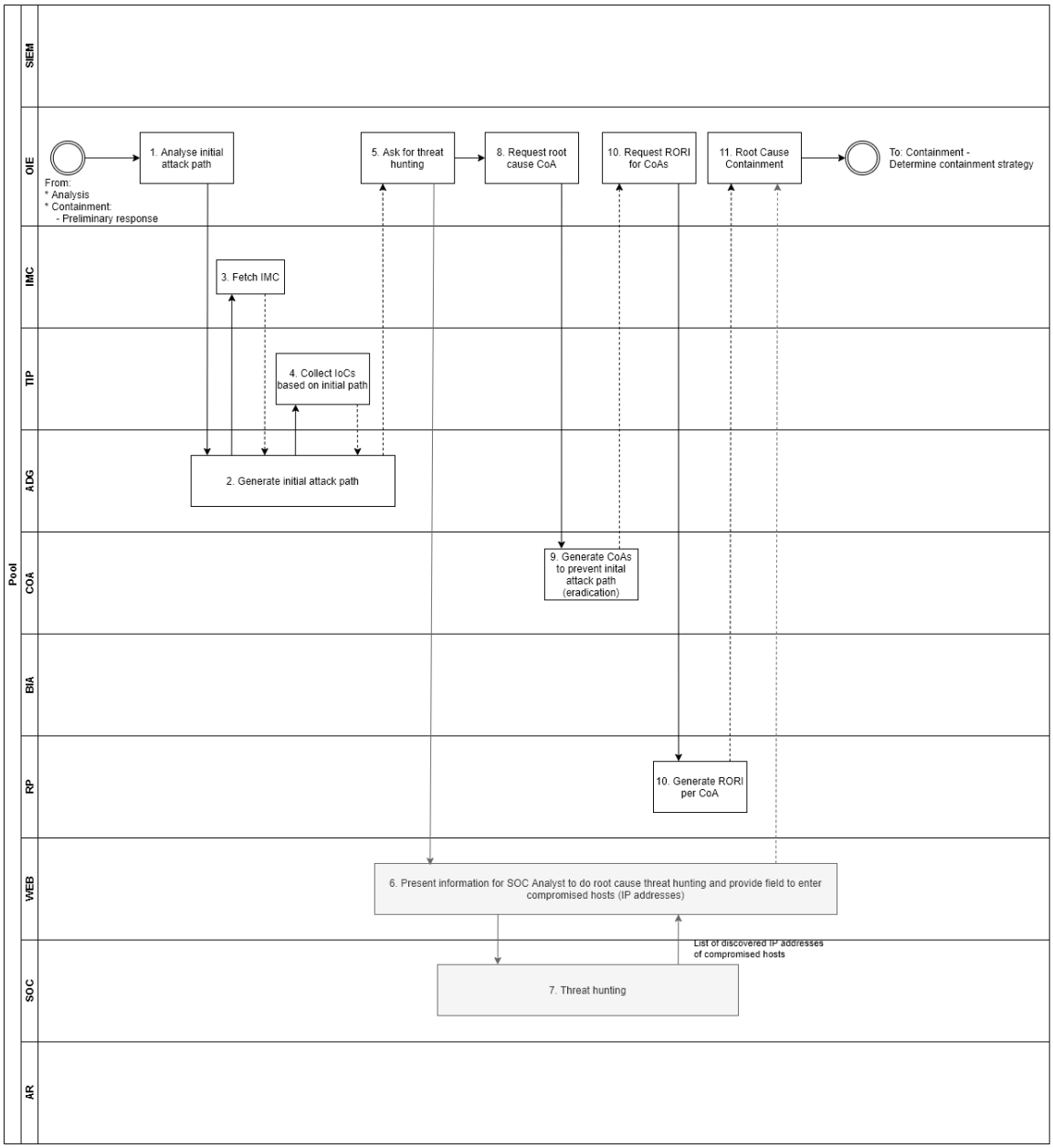


**Figure 4.7 – Swim lane diagram for assessment of full extent in containment phase of Use Case 1.**

During the detection phase, the AAD may have set the event type to either initial access, lateral movement, or exfiltration. This event type is confirmed by the SOC analyst later in the workflow, during the analysis phase. If the security incident was not defined as “initial access”, the question is how the attacker managed to get initial access, and what path the attacker used to move to the compromised asset. This can be identified by a root cause analysis (see Figure 4.8), as performed during this sub-workflow.

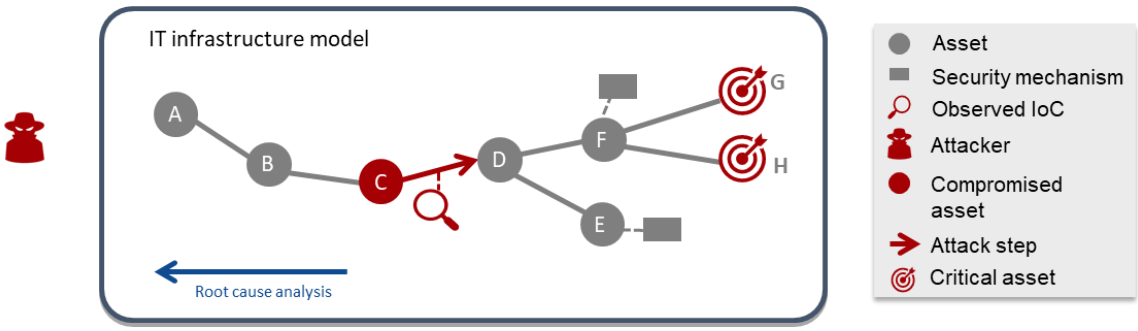


**Figure 4.8 – Conceptual visualisation of root cause analysis with the ADG.**

1. The OIE will start the workflow by querying the ADG to learn what the most probable attack path would have been through the infrastructure to reach asset C. The parameters for this ADG analysis are given in Table 4.2. This is a similar call to the ADG *simulation* function as performed for the next step analysis, but the *entryPoint* set outside the company's IT infrastructure (i.e. internet), and the compromised host, Asset C in Figure 4.8, is set as the *highValueAssets* for the simulation.

**Table 4.2 – Parameters for ADG simulation for root cause analysis.**

| Simulation Parameter | Root cause analysis |
|---|---|
| **date** | Date of successful attack (retrieved from SIEM security incident information) |
| **threatActor** | Threat actor from TIP, or default if unknown |
| **entryPoints** | Internet (default) |
| **highValueAssets** | Asset C |
| **vulnerabilities** | Default (all) if unknown |

   The date must be set to the *date* of the attack, as the IMC may have changed significantly since then. If the threat actor is known from TIP and/or SIEM information, it can be added to the simulation.
2. The ADG will fetch the infrastructure model from the IMC and will return the most probable path from outside the infrastructure to asset C, containing TTCs for the individual attack steps. In its response, it will also refer to CVEs used during the attack.
3. Using the *url*, *username* and *password* defined in the configuration of the ADG, the IMC will be retrieved and loaded by the ADG.
4. Based on the results of the ADG, the Cortex analyser may already enrich the attack steps with known IoCs by contacting the TIP. Within SOCCRATES this will be done only for attack steps that exploit a CVE. The IoCs from the TIP can guide the threat hunting team while searching for evidence of the root cause attack path.
5. The OIE receives the ADG root cause analysis results, including IoCs from the TIP and asks the threat hunting team to start the iterative process of finding compromised assets in the infrastructure with the related IoCs. The task of threat hunting can be executed in parallel.
6. The web interface will provide the relevant information (CVEs, IoCs, suspicious IP addresses, etc.) for the SOC analyst to start its threat hunt activities.
7. The threat hunter will take this information and may start adding identified additional compromised hosts on the root cause attack path or other hosts that were identified as compromised (e.g. target hosts in observed lateral movement steps). At some point the SOC analyst will determine to finish the threat hunting process via the web interface.

8. In the meantime, based on the output of the ADG, the Course of Action Generator is requested to identify the best CoA (given a certain cost) to eradicate the initial access the attacker may have used. For this, the *coagen* function (3.5.3) of the Cortex analyser of the Course of Action Generator is called with the *simulationId* returned in step 2 by the ADG. Optional monetary and time constraints costs can be provided by the SOC analyst.
9. The CoA generator returns CoAs to prevent the initial attack path, containing the efficiency of a CoA, TTC after implementing the CoA, time and monetary costs for implementing the CoA and a list of defences that can be applied. Note that these defences are based on securiCAD that may not be directly enforceable (i.e. may require extensive reconfiguration or acquisition of security products). Not directly enforceable CoAs will also be provided for post incident improvements. To evaluate the CoAs, the RORI of each CoA is requested, since a negative RORI indicates that the CoA should not be implemented.
10. To evaluate the CoAs, the RORI of each CoA is requested, since a negative RORI indicates that the CoA should not be implemented.
11. The RP requires the *siem_label*, which is set by the AAD in the detection phase and confirmed by the SOC analyst during the analysis phase. The *initialTTC*, the value of the TTC before applying any CoA, can be found per component of the CIA triad in the ADG output. Since these are not containment CoAs yet, the containment boolean should be set to False.
12. At this stage all identified compromised hosts from threat hunting are collected. The (direct enforceable) CoA will not be applied immediately, as this may alarm the attacker while the platform is still working on a complete containment strategy. Therefore, the root cause COAs will be collected and stored by the OIE, to be executed during the final containment.

### 4.2.5 Containment – Final containment

In Figure 4.9, the swim lane diagram is depicted for the final containment. If the SOC analyst did not decide to stop the workflow, all ongoing attacks from all attack stages (initial access, lateral movement, and exfiltration) pass through the final containment (sub-)workflow. The task of the final containment workflow is to decide on the optimal containment CoA, to start the containment of the compromised assets. During the execution of the entire Use Case 1 workflow, the OIE may have collected four kinds of CoAs:

1. CoAs to stop exfiltration (that at this stage have already been executed).
2. CoAs to prevent the next step from the compromised host.
3. CoAs from the root cause analysis to eradicate vulnerabilities that enable the attacker to compromise the infrastructure (generated in previous section).
4. Containment CoAs to contain compromised hosts.

In this part of the workflow the second and fourth type of CoAs are generated. The Response Planner will play an important role in generating the containment CoAs and ranking the CoAs based on RORI or Business Impact (e.g. negative effect of the CoA on the business processes), as is be shown in the swim lane diagram.

1. In order to get the full picture of what the current business impact is taking into account the additional compromised hosts found during the threat hunting activity, a BIA is requested by the OIE.
2. The *bia_analysis* function of the Cortex analyser is called by the OIE, passing multiple assets wither their *assetID*, *TTC* and *probability* for each component of the CIA triad. Since these

assets are already considered to be compromised, the TTC is set to the lowest value, and the probability to the highest value (i.e. 1) if not explicitly investigated during the threat hunting activities. The calculation can be done for the highest level of the business, so business function/process name can be left empty. The attack time should be set to the moment of the assets being compromised to get a realistic view of the business impact.

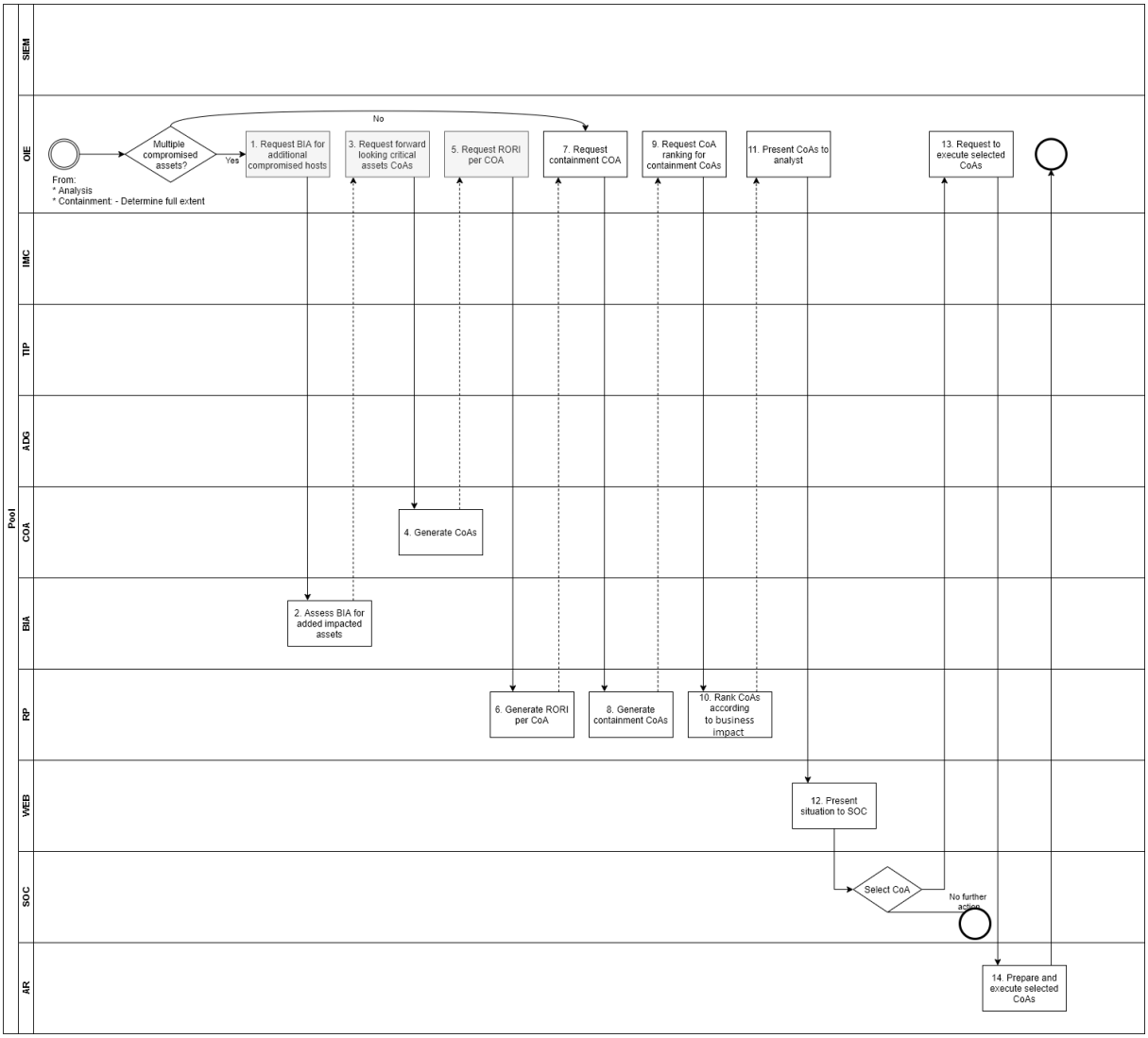


**Figure 4.9 – Swim lane diagram for the final containment in Use Case 1.**

3. The OIE stores the result of the BIA and will ask the CoA generator to generate CoAs to protect the critical assets.
4. The CoA Cortex analyser function "*coagen*" will be called, using the parameters *simulationId* to refer to the ADG simulation from the next step analysis with the ADG (see section 4.2.2), and monetary and time budgets set by the SOC analyst.
   Note: no ADG next step analysis is done for the compromised hosts provided by the Threat Hunting. This may be added to the workflow.
5. The returned result contains per CoA the efficiency, the TTC after implementation, monetary and time budget spent for the CoA and a list of defences. To evaluate the CoAs, a RORI for each CoA is requested, since a negative RORI indicates that the CoA should not be implemented.

6. The RP requires the *siem_label*, which is set by the AAD in the detection phase and confirmed by the SOC analyst during the analysis phase. The *initialTTC*, the value of the TTC before applying any CoA, can be found per component of the CIA triad in the ADG output. Since these are not containment CoAs yet, the *containment* boolean should be set to False. Finally, the CoAs returned by the CoA generator in step 4 are used as input for the *coas* array.
7. Having generated all BIAs, CoAs and their related RORIs, the OIE will now request the RP to generate the containment CoAs for all compromised assets. In Figure 4.10, the scope of a containment is depicted.

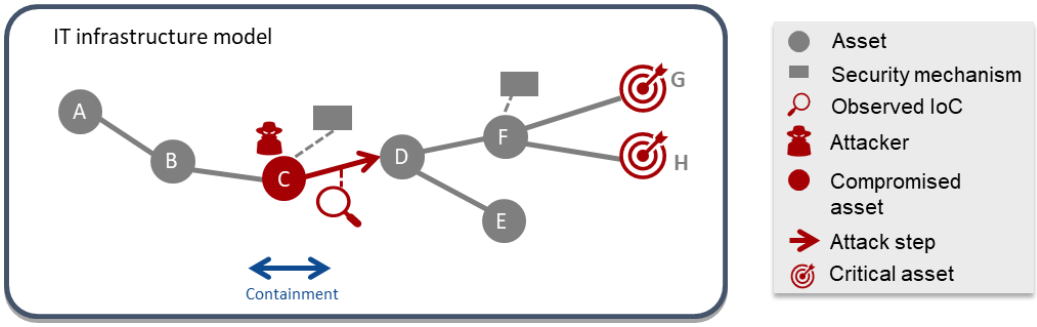


**Figure 4.10 – Conceptual visualisation of containment scope.**

8. The OIE calls the RP Cortex analyser function "generateContainmentCoas" (section 3.7.3), containing the label of the attack stage, and the IP addresses within the IT infrastructure. The RP will return the total cost and will list defences for each containment CoA.
9. The OIE will store the result of the RP. Since the SOC analyst needs to be aware whether this containment CoA could have negative business impact, the Response Planner is asked to assess business impact and rank the containment CoAs.
10. The Cortex analyser of the RP will be called to rank the CoAs, using the *coa_ranking* function. This function requires the *siem_label* (i.e. attack stage), the *containment* parameter (which is set to True), and the list of containment CoAs generated during step 6.
11. Having completed the analysis for the insight into the situation the organization's infrastructure is in, and generated appropriate CoA options and related consequences that are available, the OIE will present the CoAs to the SOC analyst as option awareness through the web interface. Note that the web interface will include information of the different types of CoAs (prevent the next step CoA, root case CoA, containment CoA). When a stop exfiltration CoA has been executed, this should also be presented to the SOC analyst.
12. The SOC analyst is asked to choose the desired CoA to be executed, that are sent to the OIE.
13. The OIE will store the selected CoA and send them to the Cortex responder of the AR component for execution.
14. As a final step of the workflow of Use Case 1, the CoAs are sent to the Cortex responders of the AR. Due to their relative complexity, the CoAs related to the next step analysis and root cause analysis will be sent to the Cortex responder responsible for preparing a ticket in the ticketing system (either by e-mail or directly through the API of the ticketing system). The (directly enforceable) containment CoAs will be specified as CACAO playbooks, containing OpenC2 commands. These OpenC2 commands can be extracted from the CACAO playbook and translated by the OpenC2 consumer into concrete commands, for instance to trigger their Endpoint Detection and Response platform (section3.10).

## 4.3 Use Case 2-5

As mentioned in section 4.1, Use Cases 2 to 5 are used in a pre-attack phase. These use cases differ in their starting point in the workflow, and by which component they are triggered. The analysis steps, however, are identical. Therefore, these use cases are described in one section. In Figure 4.11, the combined swim lane diagram is depicted for these use cases.

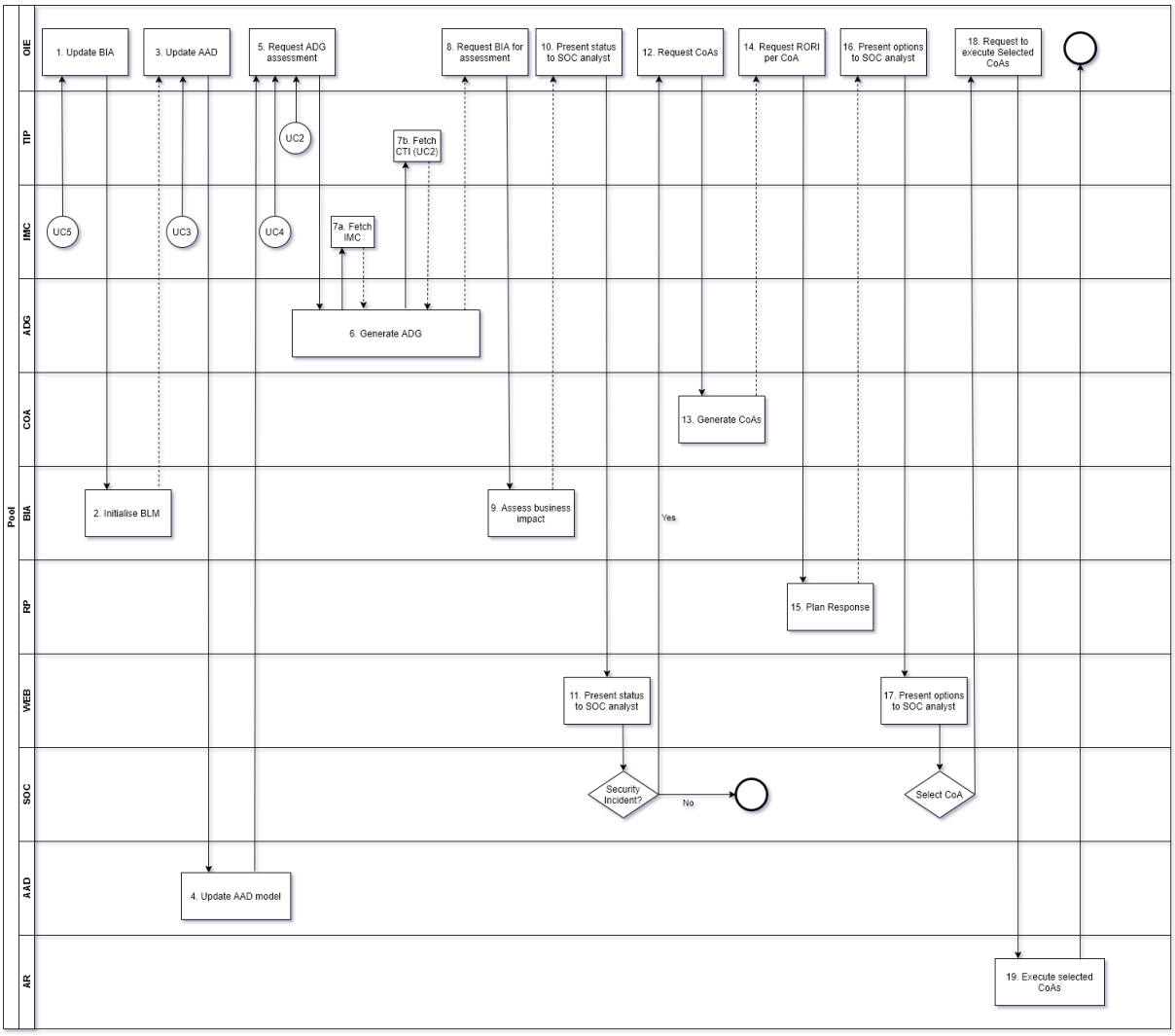


**Figure 4.11 – Swim lane diagram for Use Case 2 - 5.**

1. For Use Case 5, the IMC will trigger the orchestrator's workflow. When (part of) a new infrastructure is added to the IMC by one of the adapters, the IMC will trigger a new workflow at the OIE using the REST API described in section 3.2 as the "Specification of API endpoint that triggers a new workflow at the OIE". All newly added assets will be posted to the orchestrator's REST API.
   When new systems are added to the infrastructure, the BIA will need to update its Business Logic Model.
2. The OIE will call the Cortex responder at the BIA component, using the *bia_updateModel* function described in section 3.6.3. The orchestrator will fetch the information of the newly added assets, business process, business functions and their mappings to these assets. This information is provided by the pilot sites.

3. When a vulnerable asset was discovered (UC3) or when new assets are added to the infrastructure (UC5), the AAD component will need to be informed since REASENS need to update its detection model.
4. Depending on the use case, the Cortex responder of the AAD may expect two different POST requests as described in section 3.1.2. In case of UC5, the responder will be called with type "*newAsset*", providing the hosts. When successful, the AAD will respond to the OIE that the AAD successfully updated its model.
5. When new Cyber Threat Intelligence is received by the TIP (UC2), or when a system configuration change was discovered by the IMC (UC4), the OIE will be triggered to start a workflow at this stage in the swim lane. Based on the value in the "*message*" parameter, the OIE will know which workflow to start and what additional values it expects. This is described in section 3.2.2.
   As a first step for UC2-5 the ADG is consulted.
6. Depending on the use case, the ADG is called as described in section 3.4, using different values for its parameters are presented in Table 4.3.

**Table 4.3 - Parameters for the different ADG simulation for use case 2–5.**

| Parameter | Use Case 2 | Use Case 3 | Use Case 4 | Use Case 5 |
|---|---|---|---|---|
| **date** | Default (current) | Default (current) | Default (current) | Default (current) |
| **threatActor** | Name | Default (generic) | Default (generic) | Default (generic) |
| **entryPoints** | Default (internet) | Vulnerable asset | Changed asset | Added asset |
| **highValueAssets** | Default (use IMC) | Default (use IMC) | Default (use IMC) | Default (use IMC) |
| **vulnerabilities** | Default (all) | Vulnerable asset, including CVE | Default (all) | Default (all) |

7. The ADG uses the IMC to fetch infrastructure data, based on the *imc_url*, *imc_username* and *imc_password* that is provided in the configuration of the ADG component. If not specified differently, it will fetch the latest infrastructure model available (step 7a in Figure 4.11). Since this infrastructure model can become substantial in size, the ADG Cortex analyser queries the IMC directly and does not ask Activiti to provide it with the infrastructure model.
   As a result, the ADG produces an array of objects with a reference to an *assetID* in the IMC and *TTC* and *probability* of compromise for confidentiality, integrity, and availability.
   For Use Case 2 the ADG can fetch details on the capabilities of the threat actor from the TIP (step 7b in Figure 4.11).
8. For Use Case 3, 4 and 5, if the path returned in the result of the ADG includes (one of the) assets of the initial query, it means that the TTC has changed because of the event (new threat intel, discovery of a vulnerable asset, etc.).
   For Use Case 2, it is evaluated by the OIE whether one of the new CVEs is involved in the shortest path returned by the ADG.
   If the TTC has changed because of the event, the BIA is consulted to check the potential impact of it.
9. The BIA will use the result of the ADG to retrieve the impacted assets and their level of compromise in terms of confidentiality, integrity and availability as described in 3.6.3. For Use Cases 2-5, the BIA is computed for the highest level business entity for the current time.
   The result of the BIA consists of impact probability, criticality and critical time and is returned to the OIE.

10. The OIE keeps track of the workflow and stores all information related to it. After the ADG and BIA shared their results, the situation is presented to the SOC analyst through a web front-end.
11. The situational awareness for the current event is presented to the SOC analyst with the graphical interface of the IMC (section 3.2.2), the *report_url* provided by the ADG, and results of the BIA. The SOC analyst is then asked to assess whether this is indeed a security incident, and whether to generate courses of action to mitigate the risk.
    When the analyst decides that this is not a security incident, the workflow is stopped. Otherwise, the OIE is triggered to continue.
12. In case it was decided by the SOC analyst that there is a security incident, the OIE will request CoAs from the COA generator through its Cortex analyser, providing the ADG simulation id, and monetary and time costs that are optionally provided by the SOC analyst as described in section 3.5.2.
13. Upon request, the CoA generator will evaluate available CoAs and budgets, and will return a list of CoAs containing the efficiency, projected TTC after implementation, time and monetary cost and list of defences to be applied per CoA.
14. The OIE receives a list of CoAs as a response. To enrich this information for the SOC analyst, a RORI per CoA is requested from the Response Planner through its Cortex analyser, using the *coa_ranking* function.
15. For the RP to give the correct RORI, it needs the following information:
    - The attack phase (or event type) from the AAD.
    - The TTC before applying any CoAs.
    - The fact that these are not containment CoAs.
    - The list of CoAs from the CoA generator.

    The RP returns an array of CoAs to the OIE, accompanied by their RORI values and business impact.
16. The OIE collects all available information and presents it to the SOC analyst through the web interface.
17. The SOC analyst is shown the same information as in step 11, but now with the different CoAs that are available for the security incident, together with the RORI per CoA. This enables the analyst to select the preferred CoA and send it to the OIE.
18. The OIE receives the preferred CoA, providing the REST API of section 3.2.2. and stores the information of the security incident. Finally, it will send the selected CoA to the Automated Reconfiguration (AR) Component for execution.
19. The AR receives the CoA through its Cortex responder and will either inform an IT support administrator to enforce the CoA, or automatically reconfigure the infrastructure based on the CoA.

# 5 SOCCRATES Platform deployment

This chapter provides some guidance for the deployment of the SOCCRATES platform. First the deployment options are laid out and some considerations for the deployment of the SOCCRATES platform are presented. One major consideration is security, since the SOCCRATES platform will be able to control the IT infrastructures. A misconfiguration, or active adversary abusing the SOCCRATES platform could have serious consequences. The results of the security analysis of the SOCCRATES Platform is presented in a separate section, including recommendations for organisations that consider deploying the SOCCRATES Platform.

## 5.1 Deployment options and considerations

### 5.1.1 Deployment options

When organisations decide to set up a Security Operations Centre they are confronted with many questions. One of the main questions to be addressed is whether to run their own in-house SOC, or to outsource (parts of) the operational tasks to a Managed Security Service Provider (MSSP). Some organisations choose an in-house SOC due to the sensitive nature of the data and incidents, and due to the strategical importance of managing the cyberthreats for their business. Other organisations may choose to outsource because security monitoring and response operations require staff working 24 hours a day, 7 days a week, 365 days a year. On the other hand, some organisations may not have the expertise, nor the resources to build and maintain their own in-house SOC capability. The SOCCRATES platform was designed with the aim to support both deployment options.

In recent years there has been an increase of cloud-based security products being offered as Security as a Service (SECaaS). Moreover, security products typically used by SOCs are being provided as SECaaS. Many SOCs are, however, reluctant to deploy parts of the SOC technical infrastructure within a third-party cloud environment, as the data a SOC collects and processes is security and privacy sensitive. Nevertheless, it is expected that the number of cloud-based security services will continue to grow and that over time this may also become more common and acceptable for SOCs. Furthermore, some of the components of the SOCCRATES platform can already be provided as SECaaS. Therefore, four deployment options are distinguished for the SOCCRATES platform, as depicted in Figure 5.1.

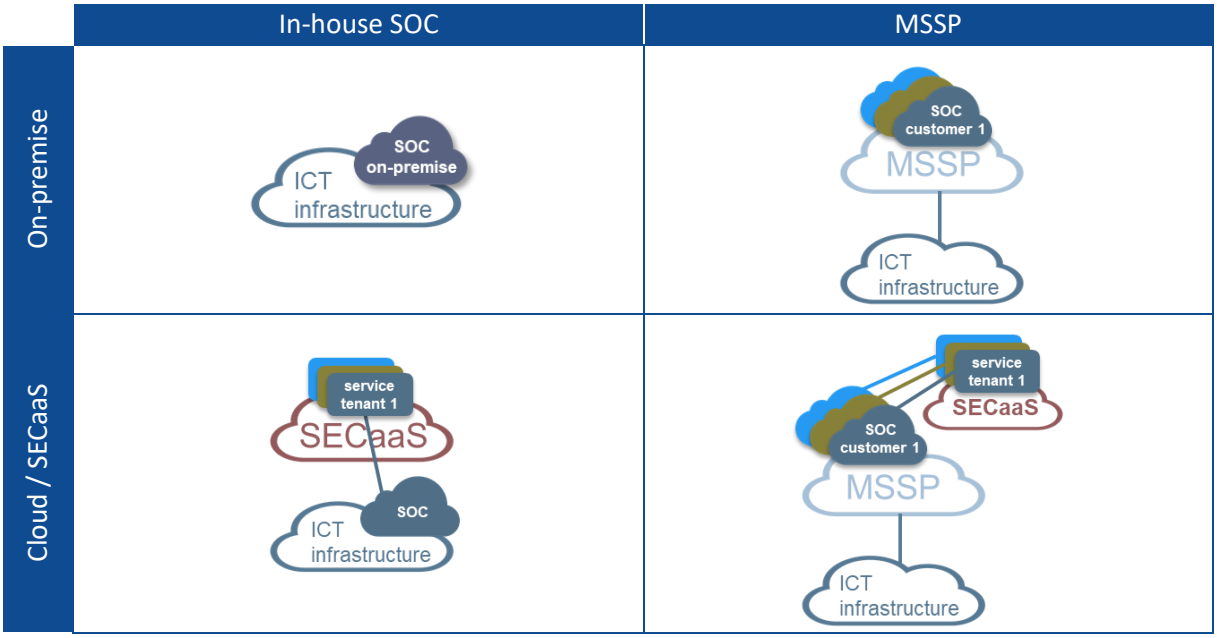


**Figure 5.1 – Four deployment options for SOC technical infrastructure.**

The SOCCRATES platform is designed for an organisation's enterprise ICT infrastructure. The SOCCRATES platform can potentially be deployed in other application areas, such as Industrial Control Systems, cloud (e.g. AWS, AZURE), Intelligent Transport Systems (e.g. Cooperative, Connected and Automated Mobility (CCAM)), and telecommunication networks. Such environments, however, will introduce new types of assets and constraints on how these systems can be managed. For example, automatically responding to a cyber security incident in Industrial Control Systems, by changing configurations or patching software, may well be much more restricted as the consequences from such operational changes may have enormous operational and societal impact (e.g. power outage). In other words, it will be necessary to assess the constraints of the environment and to ensure that SOCCRATES components can address these constraints sufficiently. Moreover, SOCCRATES components may have to be extended to handle the domain specific assets.

### 5.1.2 Deployment considerations

#### 5.1.2.1 Integration into an environment

For the deployment it is important to assess how the SOCCRATES platform can be integrated into the existing SOC environment and how to interact with the ICT infrastructure. For instance, a SOC may already have a SIEM, vulnerability scanning and management tool, threat intelligence platform and Security Orchestration, Automation and Response (SOAR) system. The SOCCRATES Platform is designed with the challenge of integrating with existing SOC environments in mind and allows for components to be exchanged, removed, or added. This will, however, require engineering and development work.

Furthermore, the SOCCRATES platform relies heavily on source data collected from the ICT infrastructure. Among others, the AAD and the IMC require data to be collected and made available. For the IMC this may again require engineering and development work as the SOCCRATES project has made only a limited set of source adapters for importing infrastructure modelling data (e.g. vulnerability scanning data, endpoint security posture, network topology, firewall rules). Unfortunately,

there is a lack of standardisation in this domain and products traditionally do not provided open interfaces to enable data to be collected and used by other (security) tools.

For the execution of CoAs in the ICT infrastructure, the SOCCRATES platform has to integrate with IT service management and APIs to reconfigure security functions. Many organisations are, however, reluctant to the concept of automated CoA execution. Therefore, the SOCCRATES platform has to integrate with IT service management, typically by automatically sending an IT support ticket. As described in section 3.10 the SOCCRATES platform can interact with some IT support ticketing systems. Regarding automated reconfiguration there is ongoing standardisation work to develop common language for requesting a security commands (i.e. OASIS OpenC2, IETF i2nsf) and playbooks (i.e. OASIS CACAO) for more complex security commands and response actions. As this is ongoing development it is not likely that all security functions in an ICT infrastructure will already support these standards. Therefore specific Cortex responders might need to be developed to interact with the security functions an organisation has.

Another aspect of integration is the need for business process and business function information for the Business Impact Assessment. Some organisations may already use Enterprise Architecture or Business Process Modelling products that can be used as a source for the BIA. If an organisation does not have this available, the organisation has to either create such information or come up with an alternative approach for the functions provided by the BIA.

Last but not least, each organisation has its own incident response processes. Although the workflows of the OIE are generic in analysing security events and incidents, an organisation has to do some configuration and tuning of the OIE and workflows to customize them to their workflows and procedures.

#### 5.1.2.2 Staffing

Based on the previous section, the SOCCRATES platform will introduce several new systems that require engineering, both during the initial setup as well as during later stages of the platform's life cycle. For instance, to improve efficiency and effectiveness, or to introduce new capabilities. This is an important issue when considering the impact on staffing when deploying the SOCCRATES platform.

The following Staffing issues need to be considered:

- *SOC / CSIRT analyst work will change* – as with all automation the role of the human operator will change. The introduction of a platform like SOCCRATES will likely also change the operator's role. In particularly when the platform's analysis results and the CoA suggestions become highly accurate and much faster. We expect that the role of the SOC / CSIRT analyst may change into two directions:
    1) the SOC / CSIRT Analyst will do more complex (threat hunting) cases that are difficult to automate; and
    2) the SOC / CSIRT Analyst may focus more on improving and tuning the platform and thus becomes a "SOC system engineer".

  It is expected that this change will happen gradually, such that an SOC organisation can anticipate these changes. For the initial deployment it is important to get people from the SOC / CSIRT team acquainted with all the different new tools and functions, to improve the overall SOC / CSIRT performance. In other words, provide training to these SOC / CSIRT analyst to use the tools of the platform.

- *Increased need for SOC system engineers* – analogue to the introduction of SIEM and SOAR products, it is important to be able to tune these SOC tools to the environment and threat landscape. This is the role of the SOC system engineer. As described above, the SOCCRATES platform introduces a variety of new functions that each require some level of tuning and improvement over time. It is necessary to have sufficient system engineering resources for the SOC / CSIRT systems.
- *SOC system maintenance* – The SOCCRATES platform introduces a whole set of new systems that need to be maintained. On the short term, it is expected that the capacity for system maintenance work will increase.

## 5.2 Security Analysis of the SOCCRATES Platform

This section provides some guidance for organisations that consider deploying the SOCCRATES platform with respect to the security of the platform.

In the literature on developing, implementing, and operating a SOC, the security of the SOC itself is not often addressed. In ENISA's *"How to setup up CSIRT and SOC"*, from December 2020 [11], CSIRT IT and information security management plan is only briefly addressing the topic and mainly references ISO/IEC 27001, COBIT and ITIL. The best guidance on security of the SOC itself is given in MITRE's *"Ten Strategies of a World-Class Cybersecurity Operations Center"* [12], in which Strategy 8 is: *Protect the SOC Mission*.

In this security analysis, we first define the scope and some assumptions. Next, we will identify and assess the assets to a SOC / CSIRT deploying the SOCCRATES platform and a high-level overview of threat vectors to SOCCRATES platform are given. The section is concluded with some recommendations.

### 5.2.1 Scope and assumptions

The scope of the security analysis is all the systems and SOCCRATES components that will be hosted by the SOC / CSIRT. With respect to the SOCCRATES component architecture that means that the Threat Data Collection & Threat Prediction component, that is hosted by Shadowserver, is out of scope of this analysis. The scope demarcation is visualised in Figure 5.2.

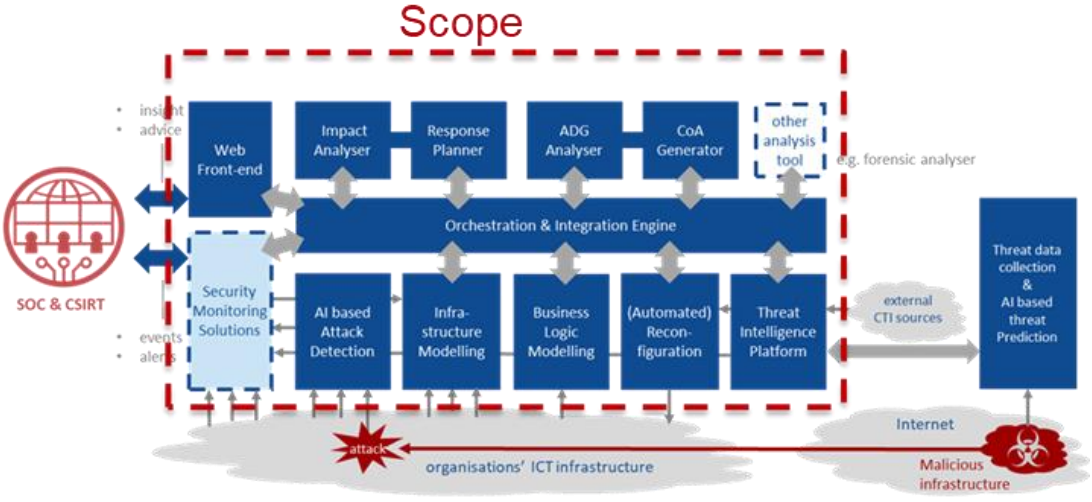


**Figure 5.2 – Security analysis scope demarcation in SOCCRATES component architecture.**

Based on among other [12] the (traditional) information security goals for a SOC can be summarized as:

A. Ensure the *availability* and *integrity* of the security monitoring source data.
B. Prevent adversaries to detect or gain access to information on the detection coverage, capabilities, and methods, as this can be used by adversaries to evade detection.
C. Make sure the SOC can operate even when the organisation's ICT infrastructure is compromised, this includes among others: communication systems, file storage systems and incident management systems.
D. Ensure *system integrity* and *availability* of the SOC system assets.
E. Ensure *confidentiality*, *integrity,* and *availability* of SOC information assets.

To achieve these security goals, it is common to apply network segment for these SOC systems (with data diodes) and own system administration and user authentication. In [12], this separate SOC network segment is referred to as the SOC enclave. We assume that a SOC will have such SOC enclave operational and will deploy the SOCCRATES platform within the SOC enclave.

For an MSSP the security goals listed above also apply, but MSSPs also have some specific security challenges: keep customer data and the access to the customer's infrastructure separate from other customers and adversaries in general.

### 5.2.2 Assets in SOCCRATES platform deployment

The SOCCRATES platform extends the capabilities of the SOC / CSIRT with new capabilities. These new capabilities also introduce some new system and information assets that need to be protect. The protection of the SOCCRATES components fall under security goal D. The new information assets introduced with SOCCRATES are listed and assess on confidentiality, integrity, and availability in Table 5.1.

**Table 5.1 - SOCCRATES platform information assets.**

| Type of SOCCRATES information assets & explanation | C | I | A |
|---|---|---|---|
| *IMC data*<br>The data stored in the IMC includes network topology, systems information (e.g. names, IP address, OS, applications), including security functions and configurations (e.g. firewall rules), current information on vulnerabilities in assets. All this information is valuable to an adversary. It thus needs to be protected against unauthorised access and manipulation. For the correct functioning of the SOCCRATES components the availability and integrity of the data is very important. | × | × | × |
| *ADG analysis results*<br>The results of the ADG can provide insight into how the organisation's ICT infrastructure can be attacked. This information can be very useful for an adversary. | × | × | |
| *Business logic models and BIA results*<br>Business logic models and information where the most important assets are can be valuable for an adversary. | × | × | |
| *Playbooks/workflows of the OIE*<br>The playbooks provide information on how the SOCCRATES platform performs the analysis. An adversary can use this information to prevent successful operations by the SOCCRATES platform against the attack. | × | × | |

| Type of SOCCRATES information assets & explanation | C | I | A |
|---|---|---|---|
| *Containment CoAs and CoA*<br>Course of Action information, particularly Containment CoAs that the SOCCRATES platform is recommending, can be useful for an adversary. An adversary can use this information to identify whether the attack is detected, thus evading containment. Moreover, an adversary with insight into the details of a workflow of the SOCCRATES platform , can also trick the system in deploying Containment CoAs to perform a denial of service attack on certain systems. | × | × | |
| *AAD model and training sets*<br>The models and training sets that the AI based Attack Detection systems uses have to be protected against unauthorized access and manipulation. An attack may try to manipulate training data or use knowledge of the models to evade detection, or trigger detection for certain systems. | × | × | |
| *Cyber Threat Intelligence*<br>Threat intelligence data and intelligence is becoming more and more important for SOC and CSIRTs. Insight into the knowledge position of the SOC / CSIRT by the adversary, or capability to manipulate the integrity and availability of CTI may provide the adversary with an advantage. | × | × | |
| *Credentials for access to SOCCRATES components*<br>The APIs of the SOCCRATES components have access control and require the consumer to have credentials. The credential thus enable access to the data and/or analysis results provided by the SOCCRATES component. | × | | |
| *Credentials for access to security function for (semi-)automated reconfiguration*<br>The security functions that the (semi-)automated reconfiguration component can control will typically require credential-based access control on its API. Access to these credentials will enable an adversary to control the security function. | × | | |

The (semi-)automated reconfiguration capability needs to be protected to prevent misuse by an adversary. An adversary may for instance try to:

- Remove certain security measures.
- Perform a denial of service attack by introducing some security measures that may have negative side effects for the operations.

The (semi-)automated reconfiguration capability and interfaces to the infrastructure need to be protected.

### 5.2.3 Threats to SOCCRATES platform deployment

As described in section 5.2.1 we assume that the SOCCRATES platform will be deployed within a SOC enclave. As described in section 5.1.1 we need to assume that (in the future) some components of the SOCCRATES platform are (partly) provided as a SECaaS. Furthermore, we assume that there can be VPN connection for SOC analysts to work from home. In Figure 5.3, a non-exhaustive high-level overview of the attack vectors to the SOCCRATES platform is depicted. The depicted attack vectors are:

1. Adversary attacks via sensor data
2. Adversary attacks via CTI data feeds and APIs (incl. collaboration platforms)
3. Adversary attacks via the reconfiguration interfaces

4. Adversary attacks via remote access solutions (e.g. VPN connection for SOC analysts from home)
5. Adversary attacks via the supply chain
6. Adversary attacks via cloud-based security services
7. Adversary attacks via insider

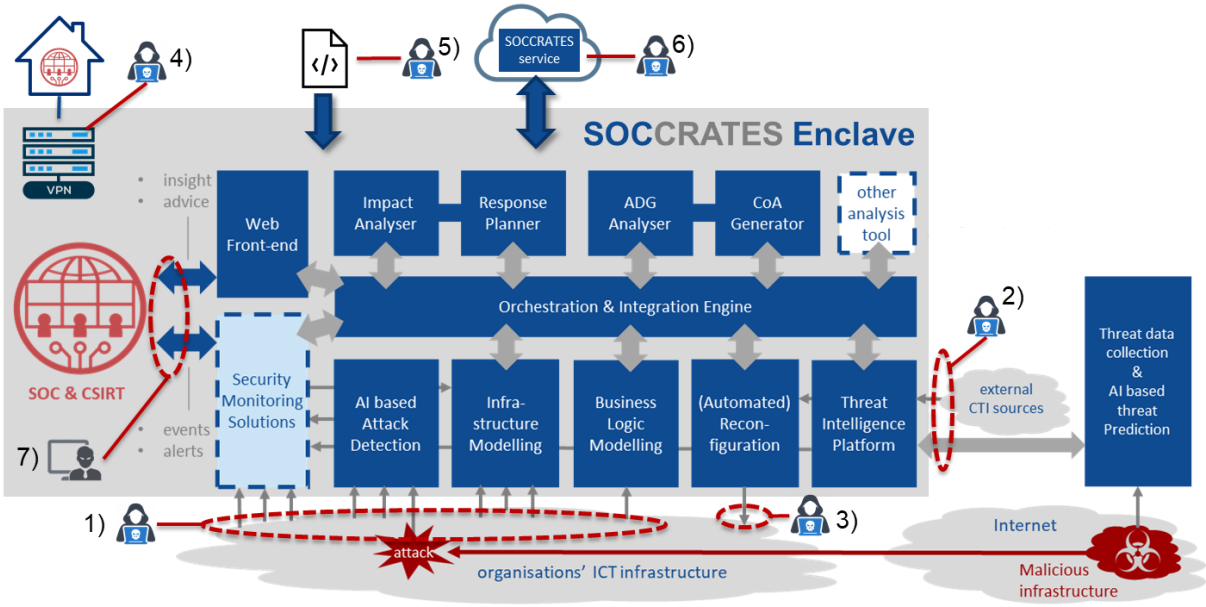


**Figure 5.3 – Overview of attack vectors to the SOCCRATES Platform.**

### 5.2.4 Recommendations

As described in this chapter, the SOCCRATES platform must be tailored to the organisation's ICT environment and integrated with existing SOC systems. The security must be assessed in this context. Therefore, it is recommended to perform an in-depth security assessment for the specific implementation within the organisation's ICT environment and SOC enclave.

The SOCCRATES platform consists of multiple software components with a wide variety of maturity, ranging from commercial products to research proof of concepts. The code quality may therefore vary. As the SOCCRATES project does not have a specific activity to perform security analysis of the software (e.g. static and dynamic application security testing) and of the APIs. In general, during the project all partners will have a best effort approach to ensure software quality and manage vulnerabilities for the proof-of-concept software used in the pilots. The SOCCRATES exploitation plan (i.e. deliverable D8.6) will contain information on the software components that are made available and how (e.g. as commercial product or as open source software) and will also include information on vulnerability management. It is recommended that organisation considering deploying SOCCRATES components, or the platform as a whole, to take this into consideration.

It is commonly recommended to apply SOC systems in a separate network segment, with among others separate system administration and user authentication, referred to as SOC enclave. It is recommended to also apply network segmentation within the SOC enclave. Create network segments for specific types of functions (e.g. for malware analysis). Other recommended security measures include:

- Security penetration testing,
- Security monitoring of own SOC systems,
- Vulnerability scanning & patch management.

Note that this is by far an exhaustive list. It is recommended that organisation follow the organisation security policy and perform control selection to mitigate the threats identified in the already mentioned security risk assessment using security best practices, such as [13], [14], [15]. Moreover, it is recommended to do this in a formal way by implement an Information Security Management System (ISMS) for the SOC / CSIRT, using ISO/IEC 27001 [16].

# 6 References


[1] Martin Eian, *"D2.1 Use cases definition & pilot sites requirements"*, SOCCRATES. December 2019.

[2] Sebastiaan Tesink, *"D2.2 System Architecture & Interface Specifications (Initial)"*, SOCCRATES, May 2020.

[3] Paul Cichonski. Tom Millar, TimGrance, and Karen Scarfone, *"Computer Security Incident Handling Guide"*, NIST Special Publication 800-61 Revision 2, August 2012.

[4] ISO/IEC 27035-1:2016, Information technology — Security techniques — Information security incident management — Part 1: Principles of incident management.

[5] An architectural blueprint for autonomic computing, IBM whitepaper, June 2005. Available at: http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf

[6] Ruben Trapero, *"D6.1 Initial version of the SOCCRATES Platform Orchestration, Reconfiguration and Front-end"*, SOCCRATES, 30/11/2020.

[7] Irina Chiscop, "D4.1 AI-based Attack Detection to Detect Advanced Threats", SOCCRATES, revision 2.0, 30/10/2020.

[8] "D3.3 Specification of the common ICT infrastructure reference meta model", SOCCRATES, 31/8/2020.

[9] Marko Komssi & Ville Alkkiomäki, *"Initial ADG based Attack prototypes"*, SOCCRATES, 28/02/2021.

[10] Christophe Kiennert, *"D5.1 Definition of the business model structure, inputs and interfaces"*, SOCCRATES, 31/8/2020.

[11] Edgars Taurins, *"How to setup up CSIRT and SOC - Good Practice Guide"*, ENISA, December 2020.

[12] Carson Zimmerman, *"Ten Strategies of a World-Class Cybersecurity Operations Center"*, MITRE, 2014.

[13] ISO/IEC 27002:2013*, *"Information security, cybersecurity and privacy protection — Information security controls"*. (new version is expected by end of 2021 or early 2022).

[14] *"Framework for Improving Critical Infrastructure Cybersecurity"*, NIST, Version 1.1, April 16, 2018.

[15] Center for Internet Security, *"CIS Controls"*, Version 8, May 2021

[16] ISO/IEC 27001:2013, "Information technology — Security techniques — Information security management systems — Requirements".

# 7 Abbreviations

| Acronym | Description |
|---|---|
| AAD | AI based Attack Detection |
| ABC | Anomaly-Based Clustering |
| ACT | semi-Automated Cyber Threat intelligence |
| ADG | Attack Defence Graph |
| AEF | Argus Event Format |
| AEP | Adversary Emulation Plans |
| AI | Artificial Intelligence |
| AIT | AIT Austrian Institute of technology |
| Aminer | logdata Anomaly Miner |
| API | Application Programming Interface |
| APT | Advance Persistent Threat |
| AR | Automated Reconfiguration |
| ATOS | ATOS Spain |
| AV | AntiVirus |
| BIA | Business Impact Analyser |
| BLM | Business Logic Modelling |
| BPMN | Business Process Model and Notation |
| CACAO | Collaborative Automated Course of Action Operations |
| CAPEX | Capital Expenditures |
| CC | Command and Control |
| CCAM | Cooperative, Connected and Automated Mobility |
| CEF | Common Event Format |
| CERT | Computer Emergency Response Team |
| CMDB | Configuration Management Database |
| COPS | Collaborative Open Playbook Standard |
| CSIRT | Computer Security Incident Response Team |
| CoA | Course of Action |
| CSV | Comma-separated values |
| CTI | Cyber Threat Intelligence |
| DC | DataCentre |
| DGA | Domain Generated Algorithm |
| DNS | Domain Name System |
| DoH | DNS over HTTPS |
| DoT | DNS over TLS |
| EDR | Endpoint Detection and Response |
| ELK | Elasticsearch/Logstash/Kibana |
| EN | Evidential Network |
| FRS | Foreseeti |
| FSC | F-Secure |
| GUI | Graphical User Interface |
| HGA | Hybrid Genetic Algorithm |
| HMAC | Hash-Based Message Authentication Code |
| ICT | Information and Communication Technology |

| | |
|---|---|
| IDS | Intrusion Detection System |
| IMC | Infrastructure Modelling Component |
| IMT | Institut Mines Télécom – Télécom SudParis |
| IoC | Indicators of Compromise |
| IP | Internet Protocol |
| IPS | Intrusion Prevention System |
| IRM | Incident Response and Management |
| ISMS | Information Security Management System |
| ITIL | Information Technology Infrastructure Library |
| JSON | JavaScript Object Notation |
| KTH | Kungliga Tekniska högskolan - Royal Institute of Technology |
| L-ADS | Atos Live Anomaly Detection System |
| LAN | Local Area Network |
| LDAP | Lightweight Directory Access Protocol |
| $M_n$ | Infrastructure Model (at time *n*) |
| MAL | Meta attack language |
| ML | Machine Learning |
| MNM | mnemonic |
| MQTT | Mosquitto |
| MSSP | Managed Security Service Provider |
| MTTD | Mean Time To Detect |
| NOC | Network Operations Centre |
| NSF | Network Security Functions |
| OpenC2 | Open Command and Control |
| OPEX | Operating Expenditures |
| OT | Operational Technology |
| OS | Operating System |
| P2P | Peer-to-peer |
| PCAP | Packet Capture |
| PoC | Proof of Concept |
| REST | Representational state transfer |
| RORI | Return on Response Investment |
| RP | Response Planner |
| RPZ | Response Policy Zone |
| SDN | Software Defined Network |
| SECaaS | Security as a Service |
| SHS | Shadowserver |
| SIEM | Security information and event management |
| SMS | Security Monitoring Solution |
| SOAR | Security Orchestration, Automation and Response |
| SOC | Security Operation Centre |
| SOCCRATES | SOC & CSIRT Response to Attacks & Threats based on attack defence graph Evaluation Systems |
| SSL | Secure Sockets Layer |
| STIX | Structured Threat Information eXpression |
| SWFE | SOCCRATES Web Front End |
| TAC | Threat Actor Context (OASIS) |

| | |
|---|---|
| TAP | Test Access Point |
| TCP | Threat Data Collection and Threat Prediction |
| TIP | Threat Intelligence platform |
| TLD | Top Level Domain |
| TLS | Transport Layer Security |
| TNO | Dutch organisation for toegepast natuurwetenschappelijk onderzoek |
| TRL | Technology Readiness Level |
| TTC | Time To Compromise |
| UC | Use Case |
| VLAN | Virtual LAN |
| VM | Virtual Machine |
| VTF | Vattenfall |
| WFE | Web Front End |
| YAML | YAML Ain't Markup Language |