D7.1
Validation Plan

The DiSIEM project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700692.
Editor

Pedro Dias Rodrigues, EDP

Contributors

Olivier Thonnard, Amadeus
Miruna-Mihaela Mironescu, Amadeus
Zayani Dabbabi, Amadeus
Gustavo Gonzalez, Atos
Mario Faiella, Atos
Cagatay Turkay, City University of London
Frances Buontempo, City University of London
Ilir Gashi, City University of London
Phong Nguyen, City University of London
Abdullahi Adamu, DigitalMR
Liliana Ribeiro, EDP
Adriano Serckumecka, FCIências.ID
Ana Respício, FCIências.ID
Pedro Ferreira, FCIências.ID
Alysson Bessani, FCIências.ID
Michael Kamp, Fraunhofer IAIS
Siming Chen, Fraunhofer IAIS

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>EDP</td>
<td>First version of the document for internal review</td>
</tr>
<tr>
<td>1.1</td>
<td>EDP</td>
<td>Document released to the coordinator for quality check and submission</td>
</tr>
<tr>
<td>1.2</td>
<td>FCIências.ID</td>
<td>Final minor improvements and submission to EC</td>
</tr>
</tbody>
</table>
Executive Summary

The DiSIEM project has defined several goals with the common ambition of extending and improving existing SIEM systems, having the following principles in mind: (1) no modification on existing SIEM systems, (2) the exploitation of existing extension ports in SIEM platforms, (3) the existence of non-integrated components, and (4) no substantial additional configuration should be needed. A reference architecture for the developments was established in deliverable D2.2, while the description and technical information regarding each component was laid out in the deliverables for work packages 3-6.

This report collects and describes the designed scenarios and developments made during the first 24 months of the project, detailing their integration and validation plan using the partners’ SIEM platforms. The components will be integrated with the SIEM systems following three different patterns – Event Generator, Event Inspector, and Event Collector – that are supported by the three platforms selected for validating the project innovations: Micro Focus ArcSight (provided by EDP), XL-SIEM (provided by ATOS) and Elasticsearch (provided by Amadeus). A detailed description of the platforms is provided in this report, noting the commonalities and differences and how they are relevant to ensure that DiSIEM developments may be integrated with multiple SIEM platforms.

The integration scope is defined in detail, using scenario-based development, considering the testbeds made available by the industry partners. The information presented here will guide the activities for the last 12 months of the project, including the definition of success criteria when integrating the developments made by the consortium partners in the provided test environments, leading to possible deployments in partners production environments.
Table of Contents

1 Introduction .......................................................................................................................... 7
  1.1 Validation Methodology ................................................................................................. 7
  1.2 DiSIEM Architecture ................................................................................................... 8
  1.3 SIEM Integration ........................................................................................................... 9
  1.4 Organization of the Document ..................................................................................... 9
2 DiSIEM Components .......................................................................................................... 11
  2.1 Multi-level Risk Manager (WP3) .................................................................................. 12
  2.2 Diversity and Forecasting Analytics Platform (WP6/WP5) ............................................. 14
     2.2.1 Diversity and Forecasting Analytics Engine (WP6) ............................................... 14
     2.2.2 Diversity and Forecasting Analytics Dashboard (WP5) ........................................ 16
  2.3 OSINT Threat Detector (WP4) ....................................................................................... 17
  2.4 Listening 24/7 Threat Analyser (WP4) ......................................................................... 18
  2.5 Threat Intelligence Integrator (WP4) ............................................................................ 20
  2.6 Network-based Anomaly Detector (WP6) ..................................................................... 21
  2.7 Skeptic II (WP6) ........................................................................................................... 23
  2.8 User Behaviour Analytics Platform (WP5) .................................................................... 25
  2.9 SLiCER Cloud Archiver (WP6) .................................................................................... 26
3 SIEM Environments ............................................................................................................ 29
  3.1 Micro Focus ArcSight (EDP) ......................................................................................... 29
  3.2 XL-SIEM (ATOS) .......................................................................................................... 30
  3.3 Elastic Stack (AMADEUS) ............................................................................................ 33
  3.4 Environment Validation ............................................................................................... 34
4 Integration Plan ................................................................................................................... 36
  4.1 Integration with SIEM .................................................................................................... 36
     4.1.1 Multi-level Risk Manager ...................................................................................... 36
     4.1.2 Diversity and Forecasting Analytics Platform ....................................................... 36
     4.1.3 Threat Intelligence Integrator ............................................................................... 36
     4.1.4 Diversity-enhanced Monitoring ........................................................................... 37
     4.1.5 SLiCER Cloud Archiver ....................................................................................... 37
     4.2 Component Integration ............................................................................................... 37
     4.2.1 Diversity and Forecasting Analytics Platform ....................................................... 38
     4.2.2 OSINT Data Fusion and Analysis Components ..................................................... 38
     4.2.3 Skeptic II .............................................................................................................. 39
5 Evaluation ............................................................................................................................ 40
  5.1 Success Criteria .............................................................................................................. 40
  5.2 SBD Validation Scenarios ............................................................................................ 41
  5.3 Ethical and Legal Aspects ............................................................................................. 45
6 Conclusions and Future Work ............................................................................................ 47
References ............................................................................................................................... 48
List of Figures

Figure 1 – DiSIEM Architecture .............................................................................................................. 8
Figure 2 – DiSIEM architecture and components .................................................................................. 11
Figure 3 – Multi-level Risk Manager component .................................................................................. 13
Figure 4 – Data flow of forecasting engine ......................................................................................... 16
Figure 5 – OSINT threat detector component architecture ................................................................. 17
Figure 6 – Threat Intelligence Integrator architecture ........................................................................... 20
Figure 7 – Network-based Anomaly Detection architecture .................................................................. 22
Figure 8 – Skeptic architecture ............................................................................................................. 24
Figure 9 – User Behaviour Analytics Platform data flow ..................................................................... 25
Figure 10 – SLiCER Cloud Archiver architecture ................................................................................ 27
Figure 11 – Atos testbed ........................................................................................................................ 32
Figure 12 – DiSIEM component interfaces ........................................................................................... 38
Figure 13 Skeptic II integration with User Behaviour Analytics Platform ............................................. 39
List of Tables

Table 1 – Options for Listening247 Threat Analyser integration
.................................................19
Table 2 – Listening247 Threat Analyser use cases
.......................................................20
Table 3 – Network-based Anomaly Detection modules
...................................................23
Table 4 – Skeptic modules
.................................................................24
Table 5 – ATOS Testbed components and requirements
.........................................................33
Table 6 – Amadeus test environment
..................................................................34
Table 7 – Amadeus production environment
..................................................................34
Table 8 – Component validation test environments
.........................................................35
Table 9 – Evaluation success criteria
.......................................................................41
Table 10 – Scenario 1 Claims
..................................................................................43
Table 11 – Scenario 2 Claims
..................................................................................45
1 Introduction

The DiSIEM project aims to improve current Security Operations Centres (SOC) by developing a set of extensions for SIEM (Security Information and Event Management) systems. These components will be validated in three test environments provided by EDP, Atos, and Amadeus. After that, a subset of them will be deployed on production environments provided by these partners for further validation.

Building upon the design and development activities carried out during the first 24 months of the DiSIEM project, this deliverable gathers relevant information regarding the implementation and validation phase. Based on the DiSIEM reference architecture and considering the SIEM operational environments from the partners, requirements and scenarios are defined in order to guide the integration of the developed components.

We adopted a scenario-based development (SBD) methodology to drive the integration activities, which is an effective way to analyse and communicate user requirements [Carroll 2000]. In the context of DiSIEM we will focus the scenarios on the SOC operation, transforming the day-to-day activities of the SOC teams in real-life scenarios for component utilization.

1.1 Validation Methodology

By using SBD, we argue that the scenario description will be as accurate as possible and the SOC teams evaluating the developed components will relate with the scenarios, since they are based on their daily activities and needs, addressing the shortcomings of existing SIEM systems. A scenario must include the setting or context description, actors, goals, actions and events. Additionally, each scenario must clearly identify the projected positive and negative effects of its application. The idea is that each scenario is easily understandable not only by the DiSIEM project team, but also by SOC users and external stakeholders such as industry personnel, development teams and researchers, thus contributing to the dissemination efforts of the project.

The methodology consists of three separate phases – Analysis, Design and Evaluation. In the analysis phase the requirements are assessed using methods such as interviews with experts, state-of-the-art study and discussion groups. The design phase consists of defining solutions, detailing their objectives, methods to achieve those objectives and user experience considerations. Lastly, the evaluation phase considers the usage of the component, firstly in a prototype and later on the final version, using that feedback to improve the development and eventually re-designing the component.

Another critical step is the target audience identification. As we mentioned before, the developed components might be used by different types of end-users, from SOC operators to C-level managers. Therefore, each scenario should be evaluated from different viewpoints in order to achieve a thorough analysis and a valid end product.
In DiSIEM we are considering 4 different points of view to assess integration scenarios:

1. **SOC operators**: end-users in the SOC teams, the primary users of SIEM systems;
2. **SIEM platform managers**: users responsible for the SIEM platform, who must consider its technological evolutions and limitations, as well as assessing the effort required to integrate the developed components;
3. **Component developers**: users responsible for component development, whose main concerns are the effort required in order to adapt the component to different SIEM platforms. These developers can also give precious feedback regarding the applicability of the components, specifically if the integration outcome corresponded to the theoretical expectations created during the component description;
4. **C-level management**: senior managers in the organization that are able to assess the relevance and quality of the developed component outputs, especially the ones related with risk management or other business-relevant Key Performance Indicators (KPI).

We believe that taking advantage of these 4 different profiles will yield more robust and useful component integration results.

### 1.2 DiSIEM Architecture

Deliverable D2.2 [D22] included a description of the reference architecture of DiSIEM and its main components, which remained mostly valid and unchanged during the component design and implementation phase (minor modifications will be explained in the next chapter). We review the high-level architecture in Figure 1, built around the SIEM integration that will be validated throughout WP7.
It is important to recall that DiSIEM defined as a cornerstone that all the improvements will be made around the basic SIEM functionality, without requiring modifications on the base system. There are four fundamental architectural principles in DiSIEM [D22].

**Principle 1:** DiSIEM components must be integrated to SIEM systems without requiring any modifications in the base system.

**Principle 2:** Extensions can be made either by feeding new events to the SIEM, by taking info from the SIEM database, and/or by creating new ways to visualise SIEM data.

**Principle 3:** Non-integrated components can be used if they operate side-by-side with the SIEM, without replicating any existing functionality.

**Principle 4:** No significant manual work should be required to setup and operate DiSIEM components.

### 1.3 SIEM Integration

All developed components will be integrated and evaluated in one or more of the partner's SIEM platforms: Micro Focus ArcSight (EDP), XL-SIEM (ATOS) and Elasticsearch (Amadeus). As explained in deliverable D2.1 [D21], the variety of SIEM platforms available and the fact that they represent some of the most relevant commercial and experimental SIEM platforms in the market is very significative. We are confident that successful validation outcomes in these four platforms proves that the interfaces are flexible and ensures that the components may be integrated with other SIEM systems. The components are divided into three types:

- **Event Generator:** components that generate events for the SIEM. For instance, new intrusion detector systems or OSINT analysers, which feed the SIEM with new information;
- **Event Inspector:** components that access the SIEM database to inspect data (e.g., assets and their properties) and subsets of collected events. This is especially important for visualisation and analysis tools;
- **Event Collector:** The component needs to have full access to events in the SIEM. An example is any side-systems that do additional processing of events, beyond what the SIEM is capable of.

### 1.4 Organization of the Document

This chapter presented a brief introduction to the aims of this document and some background material, preparing the reader for the rest of the report, which is organized as follows. Chapter 2 describes – in high level – the ten components designed and implemented in the DiSIEM project, while Chapter 3 presents the SIEM environments in which the components will be tested and integrated with. The integration plan is laid out in Chapter 4, including all the relevant interfaces.
mid components and between the components and the SIEM platforms. In Chapter 5 we address all the validation activities and how effective were the planned integrations. Concluding the document, Chapter 6 presents our final remarks and plans for future work.
2 DiSIEM Components

In this chapter, we present the components developed in the DiSIEM project. Figure 2 shows the updated project reference architecture with all the components and their relationship with other components and the SIEM.

The figure shows nine components, divided in four groups, in accordance with the type of enhanced provided by the component. The OSINT Data Fusion & Analysis group contains the components related with the acquisition, processing, and integration of OSINT data, mostly developed in WP4. The Diversity-enhanced Monitoring components comprise the enhanced sensors developed in the project, mostly in WP6. As for Infrastructure Enhancements, it is a group with a single component, also developed in WP6, which improves the capacity of the SIEM to securely archive events for long time periods. The last group, Visualisation and Analysis Tools, contains the components for visual analysis and forecast of collected SIEM data. Most of these components are created in WP5, implementing also techniques developed in WP3.

The architecture of Figure 2 updates the reference architecture described in deliverable D2.2 in three ways. First, the component names were changed to make them more mnemonic and relatable. Second, two components of the Visualisation and Analysis Tools group that could not be used separately were
merged in a new Diversity Forecasting Analytics Platform component. Third, a new component (Multi-level Risk Management) was introduced in the architecture. This component corresponds to the materialization of some results from work package 3 that could not be accurately implemented in any other components. All the components represented in this figure will be described in the next sections.

2.1 Multi-level Risk Manager (WP3)

The Multi-level Risk Manager (MRM) component is part of the visualization and analysis set of tools in Figure 2. It implements the Multi-level Risk Assessment (MRA) model proposed in deliverable D3.1 [D31] and allows one to integrate the results of the risk assessment with the SIEM. The risk assessment process of DiSIEM is asset-oriented. The MRA model considers three levels of assets: services, software applications, and hosts. Hosts support software applications, which can interact between each other, and services are composed by sets of applications. The model relies on three variables to compute the security risk score of each asset: the risk associated with vulnerabilities, the risk associated with security incidents, and the risk inherited from dependencies on other assets.

The MRM component architecture is presented in Figure 3. The central sub-component of the MRM is the model’s database. This database is populated with the information on:

- the company’s assets and the dependencies between them, including the multiple levels defined in the model (services, software applications, and hosts);
- the infrastructure vulnerabilities (discovered by a third-party software) and the vulnerabilities in the company’s software applications (discovered by pen-testing or other software vulnerabilities discovery tool); and
- the history of recent past security incidents.

To populate the model’s database with all this data, a DBImport module must be implemented. Similarly, the results of the risk assessment are exported to the SIEM by a DBExport module that extracts the information relevant and feeds the SIEM. The SIEM platform will then be programmed to differentiate alerts for events involving assets with higher risk scores to enhance its monitoring and detection capability, thus improving risk mitigation.

The MRM interacts with the database to update the risk score of the assets. All the interaction between the users and the MRM is done through a dashboard, connected to the database, that presents the risk assessment results and where it is possible to configure the model’s parameters. This dashboard will provide the ability to drill-down into the details of security risk, while also enabling analytic capabilities to the SOC team and security managers. Decision-making will be supported by an integrated view of the infrastructure security risk, including information about vulnerability severity and relevant incidents, coupled with interdependencies between assets for additional context.
On top of that, security risk analytics will facilitate the generation of enriched security reports to support C-level managers’ decision-making processes. To integrate the MRM component with a SIEM, the following sub-components must be customised:

- The DBImport module, which should be able to transform the information related with assets and the supported network, software and infrastructure vulnerabilities, and past incidents, to the database data model;
- The DBExport module to upload the model information into the SIEM. The SIEM integration is flexible to allow easy data exchange regardless of the SIEM technology in place, supporting both pull and push methods using standard formats.

The parameters of the MRA model can be fine-tuned in order to be aligned with the criteria for the risk assessment processes in the company. This adaptability is vital for the model, since a rigid approach could not fit the needs of entirely different organizations, operating in various contexts and with a diverse risk tolerance/appetite. The ability to customize the MRA model parameters will enable each company to address their own organizational needs, making the results relevant for both the operational and management teams.

After the integration process is complete, the rules in the SIEM correlation engine should be parameterized to establish the level of risk considered relevant to produce alerts in the SOC operation context. The outputs of the MRA model are then pondered as an additional source of information that enables SOC operators to focus on the most critical assets, while also having a better understanding of the risk associated with each asset. As the risk level is an inclusive concept, understood at the different organizational levels, from technical teams to
executive board members, the integration of the MRM component outputs will allow for business value perceptions to reach and influence the day-to-day activities of operations teams.

At the operational level, the effectiveness of the MRM component translates in accurately assessing the security risk of company’s assets and consequently improving the quality of SIEM monitoring and detection. The evaluation of the MRM component efficacy requires its continuous usage during a certain period of time. Throughout that period, one should compute the number of incidents that were detected by the SIEM using solely the information provided by the MRM component. Both the utility and usability of the component, especially with regard to analytic capabilities, should also be evaluated by SOC analysts and SOC managers by means of questionnaires.

At the tactical level, C-level managers will evaluate the enhancement on the quality of information concerning the security of services and software applications. Questionnaires targeting C-level managers should be used to assess if security risk awareness is improved.

2.2 Diversity and Forecasting Analytics Platform (WP6/WP5)

The Diversity and Forecasting Analytics Platform performs the diversity analysis on the event data provided by a SIEM, makes this data available for explorative analysis, implements forecasting models and makes these models available as interactive modelling elements. The component is developed as two tightly related sub-components developed in different work packages, the Diversity and Forecasting Analytics Engine described in [D62] and the Diversity and Forecasting Analytics Dashboard described in [D52].

2.2.1 Diversity and Forecasting Analytics Engine (WP6)

The Diversity and Forecasting Analytics Engine consumes data from a SIEM and uses that data for alert filtering capabilities, answering questions such as:

- How many IDS platforms alerted on the same event?
- How many events were flagged by all IDS platforms?
- How many events were missed by all IDS platforms?

More generally, the engine can specify how many events were flagged by $k$ out of the $n$ tools, for any $k$ between 0 and the total number of tools, $n$. The same questions can be applied to any tool that is an event source for the SIEM, including firewalls, antivirus software, IDS platforms, etc.

In our initial state-of-the-art study, we determined that such analysis is possible in some SIEM platforms, but not all. For example, no SIEM platform currently reports events missed by the monitoring tools and none suggests best/optimal combination of diverse sensors. Moreover, the alert filtering feature implementation is also SIEM dependent. The diversity count in Elastic processes the input data joining it into one index creating a vote field, so we can count the “vote” directly (e.g., 7 out of 9). An example script to upload data in this format will be provided in the final deliverables of the project.
The results can then be shown in a Kibana dashboard and, for XL-SIEM, the existing filters can be used directly. The alert filtering use case allows interactive visualisations and statistics over time, taking event data as inputs and outputs counts such as 1 out of \( n \), or \( k \) out of \( n \). Its use varies between SIEM platforms. As described, a script shows how to join or fuse the data for Elastic Search, so the vote can be found easily. For SIEM platforms using a relational database, such as XL-SIEM, the same results will be achieved via their existing join queries.

The engine also enables the labelling of specific events of interest, including false positives – events flagged as security risks, which in fact were not – and fits mathematical models according to event timestamps, aiding prediction of times to future events. The event labels can be stored directly in some SIEM platforms, including Elastic and XL-SIEM, while others may require additional storage. Once the events collected by a SIEM have been labelled, the alert filtering is enhanced by enabling counts of true or false positives or negatives, measurement of accuracy, sensitivity, specificity and, finally, it can suggest the optimal diverse configuration. Event labelling is not directly available in any SIEM we know of.

Combined with the alert filtering, labelled data provides the SIEM with enhanced assessment ability. A prototype GUI to mark specific events has been developed for Elastic. More general scripts to flag up specific records by the specific source IP address, etc. can be developed and an open source library DejaVu\(^1\) can be used to update Elastic documents. SIEM systems backed by a relational database can utilise the prototype Graphical User Interface (GUI) once appropriate create, read, update, delete (CRUD) queries have been written. The labelling use case allows marking of incorrectly classified events as false positive (an alert fired when nothing bad had happened) or false negative (something bad happened that the tools didn’t spot). It takes unlabelled events stored in a SIEM and stores the labels directly in the SIEM or in additional storage. The operational steps vary between SIEM systems, though they either involve using a front end or running scripts.

The forecasting feature aims to predict the time until the next specific event of interest, including false negative, false positive or specific type of attack. Such models are not widely available. \( R^2 \) does have some of the models we provide, but not all. Furthermore, we have expert knowledge and provide novel post-fitting procedures analysis of predictive quality and recalibration to improve prediction accuracy. The models take events of interest from a SIEM, labelled where required, and run as standalone executables. The forecasting results are made available as JSON and can be exported in syslog format too. This allows the results to be sent in to a SIEM for display or forwarded to the Diversity and Forecasting Analytics dashboard developed in WP5 and described in Section 2.2.2. In principle, the Engine can also be used without the Dashboard (for example, an industrial partner may call the Engine’s API directly from their SIEM system), so it can be used as a self-contained component. The display can also include details to assess the

\(^1\) [https://github.com/appbaseio/dejavu](https://github.com/appbaseio/dejavu)

\(^2\) [https://www.r-project.org](https://www.r-project.org)
predictive accuracy, using Reliability Growth Models (RGM). The process is shown in Figure 4.

The forecasting use case enables an advanced warning of potential problems, but also the assessment to see if the number of relevant failures in threat detection (e.g., failure of an IDS, or two IDS platforms working on a diverse configuration) is decreasing over time. The models use event timestamps as input and they output the predicted time until the next event, according to various models, together with the uncertainty associated with that prediction.

2.2.2 Diversity and Forecasting Analytics Dashboard (WP5)

The outputs of the engine can be analysed using a dashboard, as described in this section, forming the Diversity and Forecasting Analytics Platform, a single component that aims to provide insight into the diverse configurations of the monitoring tools and their predictive capabilities. The ultimate goal is to build solutions for SOC operators and cyber security analysts, helping them make better informed decisions during both the design of a defence strategy and during the evaluation of ongoing attacks and anomalies.

In terms of pipeline, the dashboard element receives data from the SIEM, invokes the engine element to apply modelling, visualises the model output, and offers interactive capability for the analysts to filter the data and refine the models. More specifically, three main stages of the analysis are described below.

Stage 1 – Overview of alerts

The alert data comes from the SIEM and, at this stage, no information on whether the alerts are real attacks or not is yet available. Due to this fact, the nature of the analysis is purely exploratory at this stage and has the goal of gaining an overview of how the alerts are distributed over time and over different configurations, and to identify trends and outliers. The dashboard aims to:

- Display the overall distribution of alerts broken down by time and other informative attributes such as protocols and IP addresses;
- Filter and focus on a particular time period or attribute value.

Stage 2 – Interactive exploration of diverse configurations

The dashboard invokes the engine to get labelled alerts, i.e., data on whether the alert is associated with a real attack or not. The dashboard enables analysts to:

- Manually adjust what the performance metrics are;
- Filter the alerts to focus on a particular subset (e.g., only false positives);
- Filter time periods;
- Observe changes over time and/or performance during a particular instance, such as an attack.

**Stage 3 – Analysis and evaluation of model ensembles**

This stage of the investigation involves a combination of the analysis modelling outputs with the aim of evaluating the forecasts for future potential vulnerabilities. The dashboard visualises the model output from the engine and enables analysts to:

- Visually investigate several models with their forecasts in a synoptic way;
- Visualise the uncertainty in the predictions;
- Relate the predicted models to past raw data to provide context to the predictions.

**2.3 OSINT Threat Detector (WP4)**

The OSINT Threat Detector (OTD) component, as defined in [D41], is part of the OSINT data fusion and analysis group shown in Figure 2. It is designed as a subscription-based service whose purpose is to provide organizations with timely awareness on the threat landscape regarding assets in their IT Infrastructures (ITI) of interest, as identified by the organization. Each subscriber organization may define multiple ITI for which the OTD will start collecting information from Twitter and raising alarms with Indicators of Compromise (IoC) related to existing and emerging threats.

The OTD component architecture is presented in Figure 5. There are four high-level sub-systems in the component: a tweet collector; a tweet processing unit; IoC communication channels; and a web server.

![Figure 5 – OSINT threat detector component architecture](image-url)

The tweet collector is responsible for gathering all the tweets from selected groups of accounts. The tweets will be filtered, classified and aggregated (clustered) in the processing unit, in order to form consistent groups of tweets.
that will be further processed and converted into an IoC format. In turn, the IoC discovered will be sent to the SOC by one or more channels: twitter; email; MISP; and SIEM. A web server will provide analysis dashboards and a management interface.

Regardless of the channel through which the analysts receive IoC providing a summary of the threat, these will contain a hyperlink to the analysis dashboards. The dashboards will provide complete information regarding the threat landscape identified by the system:

- which ITI are affected and, for each ITI, which assets are affected;
- which threats are currently active, and which attract more attention;
- access all the tweets related to an IoC;
- analyse historical IoC data.

The management interface has two main functions: managing user credentials and profiles; and managing the ITI details and characteristics, which includes creating or deleting both the ITI and assets belonging to it. Two profiles are envisaged to access the component web interface: service managers identified by the organization, and SOC operators. Besides accessing the analysis dashboards, the service managers will be allowed to manage the ITI. SOC operators will only have access to the analysis dashboards.

Besides integrating with the SIEM platforms and MISP, the component uses email and Twitter to target analysts directly in order to get their immediate attention. By integrating with MISP, the component may connect to threat sharing platforms or any other systems that use MISP. Integration with the SIEM can be achieved through MISP or directly using SIEM connectors.

Considering that threats are disclosed and confirmed sometime after they start affecting an ITI, the evaluation of the component efficacy and efficiency can only be conclusive after the threats are confirmed, therefore with some delay. The efficacy of the OTD component is related to the ability of discovering threats that affect the ITIs defined. This entails finding all the threats that affected an ITI during a certain period of time and compute the percentage that where discovered by the OTD. To evaluate the OTD efficiency in the considered time period, we have to assess the timeliness with which threats were discovered, the relevance of the threats (impact on the ITI), how they were prioritized and the actionability that was provided by the generated IoC.

2.4 Listening 247 Threat Analyser (WP4)

The Listening247 Threat Analyser component aggregates information from various publicly accessible sources on the World Wide Web, including Exploit-DB, NIST NVD (NIST National Vulnerability Database), boards, news, blogs, Twitter, etc. It also integrates information from the Dark Web and aggregates it with all the other sources of information. All this information is aggregated according to a schema based on the NIST NVD, which avoids re-inventing the wheel whilst keeping it familiar to the end users. The threat analyser consists of machine learning models that predict threat severity (e.g., CVSS 2.0 BaseMetric), threat
type (e.g., the Common Weakness Enumeration, or CWE, code that classifies the threat by modus operandi), and the targeted platform. This metadata generated from OSINT data by the threat analyser will give the SOC operators vital information about the exposure of the organization’s infrastructure, the nature of exposure as well as the potential impact of these vulnerabilities if exploited. Together with other information, such as the timestamp, it provides vital information that feeds the SOC team with information, allowing it to prioritize vulnerability analysis, and also quantify their degree of exposure in terms of number of threats with high, medium, and low impact for the infrastructure.

The Listening247 Threat Analyser relies on the Listening247 platform\(^3\) for OSINT data. However, it also gives the flexibility of integrating it with other OSINT sources via the REST API. The Listening247 platform gathers OSINT data based on a keyword search query that, in the case of DiSIEM, are the names of infrastructure such as software and hardware. It provides the option to choose the range of dates to gather data from and also selecting the sources of interest (such as blogs, videos, reviews, etc.) before harvesting data. The platform allows the SOC operators to enriching the harvested data using the threat analysis models. The enriched information can then be fetched directly from our Elasticsearch database via its REST API where it can be used as a feed for the SIEM. Table 1 summarises the integration options available.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elasticsearch REST API</td>
<td>Directly access the data enriched by the threat analyser from our Elasticsearch databases.</td>
</tr>
<tr>
<td>MISP IOC Messages</td>
<td>Get OSINT data in the form of MISP messages to your MISP instance from every time new data is available.</td>
</tr>
</tbody>
</table>

Table 1 – Options for Listening247 Threat Analyser integration

The information resulting from the threat analyser can be used in relevant use cases, as described in Table 2.

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Expected Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashboard</td>
<td>The dashboard allows the SOC operators to have an overview of the dimensions of exposure they have. This dashboard can be used to determine which vulnerabilities should be addressed first by exploring the highest severity threats for the critical infrastructure. This dashboard can also be used to explore trends in terms of the type of vulnerabilities, and the targeted infrastructure to make decision about budget planning (e.g., procurement of more secure hardware). The dashboard will provide the following capabilities:</td>
</tr>
</tbody>
</table>

\(^3\) [https://www.digital-mr.com/solutions/social-media-listening](https://www.digital-mr.com/solutions/social-media-listening)
- Interactively explore the time series of OSINT data to see the distribution of HIGH, MEDIUM and LOW severity threats and also see count for each of those classes of threat severities;
- Ability to interactively explore the CWE top 10 threat types;
- Ability to interactively explore the top 10 affected platforms or systems in the infrastructure;
- Ability to interactively explore specific sources of information, such as Listening247, Exploit-DB, NIST, and Darknet;
- Ability to read the OSINT data's actual text.

The MISP alerts can be used for direct integration with the SIEM platforms, enabling the real-time alert capabilities for OSINT data. Specifically, the SIEM will be able to alert the SOC operators using external events, such as a tweet about a relevant vulnerability that is exploited using remote connections. It can also contribute to the risk score calculations, using the vulnerability impact classification.

### 2.5 Threat Intelligence Integrator (WP4)

The scope of the Threat Intelligence Integrator is to correlate both static and real-time information, such as Indicators of Compromise (IoC), associated to the monitored infrastructure, with data coming from external OSINT data fusion and analysis tools, to infer if the latter could represent real intelligence for the SIEM. The result of this process will be the generation of a Threat Score, computed considering various criteria related to threat intelligence domain (e.g., relevance, accuracy, timeliness, variety, completeness). This additional data will be used for enriching the incoming OSINT information, before sending it to the SIEM, allowing the prioritization of received cybersecurity events.
The architecture, depicted in Figure 6, is composed of two main modules: (i) a MISP Instance, used to gather data from OSINT-based components and the monitored infrastructures. The output will be an enriched IoC sent directly to the SIEM system; and (ii) the Heuristic Module, in charge of computing a threat score, through a heuristic analysis, enriching the original data coming from OSINT data fusion and analysis tools, and stored in the MISP instance.

The integration between the SIEM and the Threat Intelligence Integrator will rely on MISP. The objective is to use, as much as possible, the built-in sharing capabilities of the platform when this interaction takes place, such as the zeroMQ⁴ publish-subscribe model. However, MISP comes with so-called “MISP-modules”, used both for ad-hoc import and export of threat information, which represent another possible solution. Moreover, new modules could be created from scratch and integrated with it without modifying the core functionalities of the platform, if needed. The details for SIEM integration are described in Chapter 4.

The final aim of this component is to enrich the incoming IoC from OSINT data fusion and analysis tools with a threat score, which represents the result of a heuristic analysis, performed correlating this data with static and dynamic information detected by the monitored infrastructure. The refined IoC will be, then, sent to the related SIEM, allowing SOC operators to prioritize incoming events. This represents also the expected outcome of the final integration.

The component will be evaluated considering valid datasets from each monitored infrastructure, used for performing the correlation with real IoC coming from the OSINT data fusion and analysis tools. However, for making the first tests, simplified datasets could also be used, for evaluating the first results given by the Heuristic Engine.

2.6 Network-based Anomaly Detector (WP6)

The Network-based Anomaly Detector component monitors and assesses network traffic, in real time, by using machine learning algorithms to detect anomalous behaviours in the monitored applications. It considers several conditions, like the number of connections between different servers, the connections and traffic between the ports used by the applications, etc. to define the boundaries of legitimate traffic and make the predictions about anomalous and potentially malicious traffic.

The outcome of the integration with a SIEM is the timely detection of anomalies or significant deviation in the behaviour of studied applications (e.g., Gitlab and OwnCloud), making it possible to detect malicious activities before they generate a security incident.

This integration will be accomplished through a text file with the results from the machine learning analysis. Such file is made available for the SIEM, so that it can be read and processed, making it possible to show in the SIEM dashboard the

---

⁴ http://zeromq.org/
comparison of the studied machine learning algorithms in terms of performance, time, accuracy, and successful predictions.

For evaluation purposes, the machine learning algorithms must be tested against a valid dataset, containing both legitimate and malicious traffic. During the training and analysis process, only legitimate traffic must be considered. Whereas during the prediction process, both legitimate and anomalous traffic is used to evaluate the algorithms based on parameters such as accuracy, precision, and recall.

Figure 7 depicts the different modules of the Network-based Anomaly Detection component and its interaction with both the SIEM platform and the monitored applications. Table 3 lists each module and their roles. More details on the component and its modules can be found in [D61].
## Table 3 – Network-based Anomaly Detection modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traffic Header Capture</strong></td>
<td>Dataset instances to be analysed during the training and prediction process.</td>
</tr>
<tr>
<td><strong>EntryPoint</strong></td>
<td>Web service where the traffic coming from the monitored applications is identified by their IP addresses and port numbers</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td>Allows interaction between the application and the end user.</td>
</tr>
<tr>
<td><strong>Training and Prediction</strong></td>
<td>Uses machine learning to train the model and make predictions over the capture data</td>
</tr>
<tr>
<td><strong>Database (DB)</strong></td>
<td>Stores the traffic capture and the output of the training and prediction process.</td>
</tr>
<tr>
<td><strong>Prediction Service</strong></td>
<td>Provides the output of the component and interacts with SIEMs and visualization tools</td>
</tr>
</tbody>
</table>

### 2.7  Skeptic II (WP6)

The Skeptic extension is an application-based anomaly detector that aims to enhance application security by monitoring user activities. The system uses a hybrid detection approach that combines UBA (User Behaviour Analytics) and Rule-based anomaly detection. While the UBA approach relies on modelling user activity patterns and detecting activities that deviate from the learnt models, the Rule-based approach uses rules defined by application experts to detect anomalies. Combining both approaches allows this component to improve detection accuracy.

Skeptic has a modular and flexible design that allows the component to monitor different application use cases and integrate with different SIEM platforms. Figure 8 depicts the different Skeptic modules and its interaction with the SIEM platform and the application monitored, as described in [D22]. Table 4 lists the Skeptic components and their roles.

The Skeptic component is an Event Generator based on the Apache Spark (version 2.1.0) and the Apache Hadoop (version 2.7.0) frameworks. For the integration with SIEM platforms and other DiSIEM extensions, Skeptic supports JSON, LEEF and SYSLOG data formats.

The Skeptic hardware requirements depend on the volume of events generated by the application to monitor. The hardware resources to allocate will be measured and fine-tuned during the integration phase. As an example, ongoing implementations of Skeptic to detect malicious scraping activities on a web application with 5 transactions per second show that, when running in batch mode, Skeptic takes around 10 minutes to analyse all HTTP traffic for a day.
Figure 8 – Skeptic architecture

<table>
<thead>
<tr>
<th>Module</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>Ingest Application Logs and OSINT Relevant Threat Intelligence data feeds</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td>Define UBA models, Rules, Scoring, Aggregation</td>
</tr>
<tr>
<td><strong>Aggregation</strong></td>
<td>Aggregate and normalize application and OSINT data</td>
</tr>
<tr>
<td><strong>Behavioural Engine</strong></td>
<td>Train detection UBA models and detect anomalous application user activities</td>
</tr>
<tr>
<td><strong>Rule Engine</strong></td>
<td>Load/Apply defined detection rules</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>Aggregate detection results and write application sessions to the SIEM</td>
</tr>
</tbody>
</table>

Table 4 – Skeptic modules
2.8 User Behaviour Analytics Platform (WPS)

The user behaviour analytics platform is an integrated system of three major DiSIEM contributions, the Visual User Behaviour Analysis (named U4-VASABI), the User Behaviour Modelling, and the Topic Modelling Based User Behaviour Extraction and Clustering. The platform reads user actions organized in sessions directly from the SIEM data, processes it and provides tools for SIEM operators to analyse this behaviour in detail. The module organization and data flows are represented in Figure 9.

![Figure 9 – User Behaviour Analytics Platform data flow.](image)

The Topic Modelling Based User Behaviour Extraction and Clustering module enables analysts to interactively understand user behaviour clusters based on ensembles of topic modelling results. Taking advantage of the LDA (Latent Dirichlet Allocation) process from text mining, we map the sessions as documents and each action in a session as a word. Thus, we can generate the topic, i.e., the probability clustered results of user behaviours. We provide a visual interface showing the distribution of initial behaviour clusters, the distribution of actions in each cluster, and the overlap among selected clusters. Analysts can understand the behaviour and interactively select suitable clusters by observing how representative they are and how much overlap exist among them. Through such an iterative manner, the analysts are able to better understand the behaviour, generate the behaviour clusters, and then further improve the user behaviour modelling. The cluster-based behavioural model provides scores for user behaviour based on predictions using state-of-the-art machine learning techniques. These scores indicate the anomaly of a user action with respect to (i) the general behaviour of users in the system, (ii) the past actions of that specific user, and (iii) the behaviour of similar users. These scores are used to annotate actions within the U4-VASABI module.

The User Behaviour Modelling module uses long-short-term-memory (LSTM) recurrent neural networks trained on user actions and the timing of these actions.
The LSTM provides a probability distribution over future actions of that user given her current behaviour. Contrasting these probabilities to the actually observed actions allows one to deduce an anomaly score of the observed actions. This anomaly score can be used to flag user behaviour directly for a SIEM operator or to use it for an in-depth analysis of a user’s behaviour. In order to obtain models for users that behave similarly, a clustering of user session is obtained from the Topic Modelling Based User Behaviour Extraction and Clustering module.

The U4-VASABI module is the part of the platform where the interactive visual analysis of individual sessions and the users are visually investigated in-depth. These solutions are designed to address a number of high-level goals and analysis tasks that are carried out by SOC operators and experts who make decisions on whether particular sessions are fraudulent or whether particular users are behaving in an unusual manner. The elements in U4-VASABI provide (i) a multi-perspective exploration of sessions through the extraction and exploration of session-level metrics and the temporal distribution, (ii) a high-level semantic summaries of action sequences through the extraction of frequent activities through pattern mining and the visual summaries of the mined patterns, (iii) a multi-level visualisation approach to investigate multiple sessions revealing their activity type and temporal distributions, (iv) a comparative analysis capability through the utilisation of expectedness scores, (v) a visual user profile to both summarise the historical behavioural profile of users and to put individual sessions in context, and (vi) a cluster based summarisation of actions to identify underlying tasks. To be able to deliver these functionalities, the U4-VASABI component involves a combination of computational methods and several interactive views. The views are coordinated within each other using a novel, multi-semantic linking approach and are provided as standalone web-based interfaces.

All the components within the User Behaviour Analytics platform are designed and built as platform independent, self-contained, generic solutions that can work with any time-stamped sequences of typed events that are organised into sessions and executed by identifiable agents (i.e., users or automated processes/agents with a given ID). All the components operate on such session-level data and are capable of accessing such data either as static files or through calls to a data store, such as Elasticsearch or any other database. The component is also capable of writing additional/computed information to these data stores. This flexibility will enable the component to be able to work both independently and in-coordination with any SIEM that uses standard data stores such as Elasticsearch.

### 2.9 SLiCER Cloud Archiver (WP6)

The purpose of the SLiCER Cloud Archiver component is to provide long-term storage for events generated by sensors in the monitored infrastructure. These events are sent to SIEM, but they stay in the system for a limited amount of time due to storage constraints (generally, less than 6 months). This component aims to use cloud storage services such as Amazon S3\(^5\) and Azure Blob\(^6\) Storage to

---

\(^5\) [https://aws.amazon.com/s3](https://aws.amazon.com/s3)

\(^6\) [https://azure.microsoft.com/en-us/services/storage/blobs](https://azure.microsoft.com/en-us/services/storage/blobs)
securely archive these events for a longer period. Increasing the retention period is important because many threats, vulnerabilities, and zero-days are discovered long after they enter the infrastructure. Some zero-day vulnerabilities take 3 years or more to be disclosed [Ablon and Bogart, 2017]. Another important reason is to support long-term forensic analysis, which requires access to events collected at the time the incident occurred.

The component organizes and stores the events in a cloud-of-clouds [Bessani et al. 2013, Oliveira et al. 2016], ensuring security, cloud fault tolerance, and cost efficiency. More concretely, the component was built to work on top of the VAWLT multi-cloud storage service,\(^7\) being developed by the FCIências.ID team. The clouds used for storing the events should be compliant with the EU General Data Protection Regulation (GDPR)\(^8\) and any additional legislation applicable to the personal or sensitive data. In the end, the objective is to eliminate the need for SIEM platforms to have a local archival infrastructure.

Figure 10 illustrates how the proposed component interacts with existing SIEM systems. The Sensors (S) generate events and send them to Connectors, which normalize and forward them to SIEM. The events may be uploaded directly from the Connectors to the SLiCER (red arrows) at the same time that they are also sent to the SIEM. However, this requires the Connectors to have direct Internet access, which might not be possible or desired in the infrastructure. The second option is to have only the SIEM uploading the information to SLiCER (blue arrow). SLiCER receives the events, groups them in time intervals (e.g., an hour), and creates an event block that is written to the clouds using the VAWLT service. To minimize the costs of using cloud storage services, it is important that the component organizes the events in such a way that typical queries on the event archive avoid downloading high volumes of data. This requirement comes from the fact that download bandwidth is the most expensive resource on typical cloud storage services. Therefore, we employ a strategy to efficiently divide event types into blocks to minimize reads from the cloud and maximize the efficiency of cloud data retrieval. This is done by generating small index files for the blocks considering event properties normally used for performing searches.

---

\(^7\) https://vawlt.io

\(^8\) https://ec.europa.eu/info/law/law-topic/data-protection_en
Archival queries are executed through a GUI provided by the component. Read event blocks are cached to be read locally, as such queries are usually part of a group of many operations, most commonly used during forensics analysis. More details about this component can be found in deliverable D6.1 [D61].
3 SIEM Environments

This chapter describes the SIEM environments of each partner, where the developed components will be integrated. We address the operational context of each SIEM since a production environment with real data and day-to-day usage has radically different implementation requirements than a test environment. The technical limitations of each platform are clearly stated, as well as the strategy to overcome those limitations and successfully integrate the components.

3.1 Micro Focus ArcSight (EDP)

The Micro Focus ArcSight, formerly HP ArcSight, is the SIEM system used at EDP by its SOC team. It is the core technical system around which the SOC operation is organised and also the main security event repository for EDP. The ArcSight SIEM platform deployed at EDP includes the full SIEM stack, consisting of three major components of its architecture: Connectors, Loggers, and the Enterprise Security Management (ESM).

The ESM component, also known as the console or correlation engine, provides a graphical interface for the SOC team. This module is responsible for event correlation, processing real-time rules and triggering alerts. From this interface the SOC team monitors, analyses, and deals with collected data, creating custom filters and rules. All the real-time events are constantly being updated and presented in dashboards, with several visualization options, such as charts, tables or network graphs.

The connectors are software components which collect events from sources, either through pull or push methods, normalizing them into a standard format – Common Event Format (CEF). The connectors are also responsible for filtering, aggregating and categorizing the events, which are then sent, in parallel, to the Logger and ESM components. This pre-processing is crucial to save bandwidth and improve the efficiency of the event collection process.

The ArcSight Logger is a universal log management solution with the purpose of optimization. The extreme high event throughput, efficiency in the long-term storage and the rapid data analysis are the main benefits of using the Logger. Moreover, the Logger serves as a secure event storage for forensic analysis and, if need be, to present to authorities in case of legal procedures.

EDP maintains two SIEM instances: the production environment on which the SOC operators work and the test environment, where new connectors and components are evaluated and assessed before being deployed to the production environment. Both environments are based on exactly the same product and version, making it possible to have a meaningful testbed. The differences between the test and production environment are the number of connectors and the computational resources available, both of which are greater in the production environment.

All component integrations in the course of the DiSIEM project will be addressed first in the test environment and, if the tests are successful, they will then be deployed in the production environment. The criteria to determine whether a
component is allowed to be implemented in the production environment is divided into functional, technical and procedural vectors. First and foremost, the developed components must be aligned with EDP security policies, legal and regulatory obligations and technical specifications. The developments must be validated by EDP’s security team, including a vulnerability assessment and load tests for quality assurance. Lastly, the introduction of the new components must not have a negative impact on the SOC operation, following the basic principles of DiSIEM that state minimal operational and integration overhead as the most relevant drivers for the project.

3.2 XL-SIEM (ATOS)

The Cross-Layer Security Information and Event Management tool (XL-SIEM) is an enhanced security data analytics platform with a high-performance correlation engine able to raise alarms from a business perspective, considering events collected at different abstraction layers.

The XL-SIEM is not a commercial SIEM like the Micro Focus ArcSight used by EDP, nor an open source platform like Elasticsearch used by AMADEUS. Instead, the XL-SIEM is a platform for research and development of new features and functionalities on SIEM environments. It is used to test and evaluate internal developments performed within Atos, providing great benefit, since the whole environment is controlled and developed internally at the Atos cybersecurity lab. The system is composed by a set of distributed agents, responsible for the event collection, normalization and transfer processes; an engine, responsible for the filtering, aggregation, and correlation of the events collected by the agents, as well as the generation of alarms; a database, responsible of the data storage; and a dashboard, responsible for the data visualization in the web graphical interface.

The XL-SIEM Agent is the resource layer component responsible for the event collection, normalization and transfer to the XL-SIEM Engine for its processing. Events are generated by different sensors (e.g., network traffic, honeypot events, intrusion detection systems) deployed on the customer’s monitored infrastructure.

The XL-SIEM Engine is the component on the back-end layer of the monitoring architecture responsible for the analysis and processing of the events collected by the XL-SIEM Agents, and the generation of alarms based on a predefined set of correlation rules or security directives. XL-SIEM is implemented in a Storm9 topology running in an Apache Storm cluster10 which allows it to take advantage of the scalability benefit of this architecture.

The Correlation Engine is the core of the XL-SIEM engine and integrates the open source high-performance correlation engine Esper.11 The correlator uses Event Processing Language12 (EPL), which allows a flexible definition of complex correlation rules. It is a SQL-like language that includes for example the detection

9 http://storm.apache.org/index.html
10 http://storm.apache.org/releases/2.0.0-SNAPSHOT/Setting-up-a-Storm-cluster.html
11 http://www.espertech.com/esper
12 https://docs.oracle.com/cd/E13157\01/wlevs/docs30/epl\guide/overview.html
of patterns, the definition of data windows or the aggregation and filtering of incoming events into more complex events.

The XL-SIEM database layer takes advantage of the OSSIM platform and storage capabilities. Moreover, it includes the capability to store both events (gathered by the agents) and alarms (generated by the server) in an external data warehouse using a RabbitMQ server. The format supported to send events and alarms in this case is JSON.

The XL-SIEM web graphical interface integrates the following visualization capabilities: (i) different graphical charts are shown to the user as an overview of the monitored system status; (ii) alarms, security events and raw logs can be visualized by the user; (iii) additional information provided by open source tools (e.g., Netflow traffic detected by Fprobe, vulnerabilities detected by Nessus or OpenVAS) is integrated in the graphical interface; (iv) it is possible to generate PDF reports with a summary of the SIEM analysis.

Atos provides a test environment composed by an instance of the XL-SIEM Agent (disiem-agent), the XL-SIEM engine, the XL-SIEM database and dashboard. There is no production environment defined in Atos for the DiSIEM project. All research and development work are performed in the test environment, where new connectors and components are evaluated and assessed.

Atos have setup a testbed composed of two Intrusion Detection Systems (IDS) – Snort and Suricata; two applications – OwnCloud and Gitlab; one XL-SIEM agent; and the XL-SIEM engine aiming at capturing the traffic coming from internal and external sources and labelling it as normal or anomalous, based on the source IP. OwnCloud, Gitlab, and the XL-SIEM Agent have been installed in separate Virtual Machines, each one containing both Snort v2.0.6 and Suricata v3.2.4 agents. The two applications and the XL-SIEM agent are hosted in the same network, whereas the XL-SIEM engine is isolated in a different network protected by a firewall, as shown in Figure 11.
The snort instance used to monitor the OwnCloud server was configured with only community.rules downloaded from the Snort site with all the rules uncommented. To monitor the GitLab server we have additionally included the registered snort rules files (as they are downloaded by default, with some of them commented).

The internal subnet is composed of company users that access OwnCloud and Gitlab applications regularly and whose behaviour is considered normal as long as they connect from work. Any connection performed by an outsider (a network origin outside of the organization network) is considered to be malicious.

Table 5 summarizes the platform and software requirements for the Atos testbed.

<table>
<thead>
<tr>
<th>Component</th>
<th>Base software</th>
<th>Additional requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DiSIEM-1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OwnCloud Server</strong></td>
<td>• Ubuntu</td>
<td>• Basic application simulating login page</td>
</tr>
<tr>
<td></td>
<td>• OwnCloud server v.10.0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• OSSEC agent</td>
<td>• Several users with different privileges</td>
</tr>
<tr>
<td></td>
<td>• Snort v.2.0.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Suricata v.3.2.4</td>
<td></td>
</tr>
<tr>
<td><strong>DiSIEM-2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GitLab Server</strong></td>
<td>• Ubuntu</td>
<td>• Basic application simulating login page</td>
</tr>
<tr>
<td></td>
<td>• GitLab v.10.2.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• OSSEC agent</td>
<td>• Several users with different privileges</td>
</tr>
<tr>
<td></td>
<td>• Snort v.2.0.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Suricata v.3.2.4</td>
<td></td>
</tr>
</tbody>
</table>
3.3 Elastic Stack (AMADEUS)

Elastic Stack is a group of open source products by the Elastic company, designed to take data from any type of source, in any format and search, analyse, and visualize that data in near-real time. As mentioned in [D21], the Elastic Stack is not a SIEM platform, but offers most of SIEM capabilities with greater flexibility, owing in part to its open source nature.

Elasticsearch is an open-source, broadly-distributable, readily-scalable, enterprise-grade search engine. Accessible through an extensive and elaborate API, Elasticsearch can power extremely fast searches that support your data discovery applications.

The Elastic Stack is used by Amadeus’ Application SOC team to monitor user activities on different Amadeus applications. To integrate a new extension to the Elastic Stack platform, an administrator has different options, depending on the type of the extension:

- **Elasticsearch REST API**: The API is highly flexible and allows CRUD operations, searches, aggregations, data and platform management;
- **Elastic Stack data ingestion modules**: Logstash, Beats Event Generators;
- **Native integration** with Elastic Stack components by creating custom plugins. Most Elastic Stack components support custom plugins: Logstash, Kibana, etc.

Two Elastic Stack Platforms are used by the SOC team: a production platform and a test platform. The SOC team at Amadeus is responsible from the implementation and validation of the DiSIEM components deployed in the Elastic Stack platform.
As with other developments, all components must be initially deployed in the test environment and only then transferred to the production environment, if the validation is successful. An extension is validated and can be transferred to production, if it has no performance issues and brings the added value expected to application monitoring.

In addition to the Elastic Stack platforms, Amadeus has General Purpose platforms for test and production environments, used to run tools interacting with the SIEM platform without being natively integrated. The tables below summarize the details of both test and production environments:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Machines</th>
<th>OS</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose</td>
<td>2 VM</td>
<td>Ubuntu 16.06, 100 GB Disk, 16GB RAM, 4CPU</td>
<td>Python 2.7/3, Java 8u171</td>
</tr>
<tr>
<td>Hadoop/Spark</td>
<td>3 VM</td>
<td>Ubuntu 16.06, 100 GB Disk, 32GB RAM, 8CPU</td>
<td>Spark 2.1.0, Hadoop 2.7.0, Java 8u171</td>
</tr>
<tr>
<td>SIEM</td>
<td>5 VM</td>
<td>Ubuntu 16.04, 100 GB Disk, 16GB RAM, 4CPU</td>
<td>Elastic Stack</td>
</tr>
</tbody>
</table>

Table 6 – Amadeus test environment

<table>
<thead>
<tr>
<th>Platform</th>
<th>Machines</th>
<th>OS</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose</td>
<td>4 VM</td>
<td>Ubuntu 16.06, 150 GB Disk, 16GB RAM, 8CPU</td>
<td>Python 2.7/3, Java 8u171</td>
</tr>
<tr>
<td>Hadoop/Spark</td>
<td>6 VM</td>
<td>Red Hat Enterprise Linux 7, 2TB Disk, 64GB RAM, 32CPU</td>
<td>Spark 2.1.0, Hadoop 2.7.0, Java 8u171</td>
</tr>
<tr>
<td>SIEM</td>
<td>10 VM</td>
<td>Red Hat Enterprise Linux 7, 5TB Disk, 128GB RAM, 64CPU</td>
<td>Elastic Stack</td>
</tr>
</tbody>
</table>

Table 7 – Amadeus production environment

### 3.4 Environment Validation

The component integration with the 3 available SIEM environments will be conducted using a sampling approach, as planned, since the complete integration and validation of all components in all SIEM platforms would require an amount of resources not available in the project. By making sure that most components are integrated with at least two SIEM platforms, and considering the fundamental differences between all SIEM environments, one can effectively demonstrate the ability to deploy the component in a flexible manner. Table 8 summarizes our planned integration for component validation. Throughout the validation phase some adjustments may be made if integration issues occur or if the partners decide that the component must be validated in additional environments.
<table>
<thead>
<tr>
<th>Component</th>
<th>Test Environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-level Risk Manager</td>
<td>Amadeus, EDP</td>
</tr>
<tr>
<td>Diversity and Forecasting Analytics Platform</td>
<td>Amadeus, Atos</td>
</tr>
<tr>
<td>OSINT Threat Detector</td>
<td>Amadeus, Atos, EDP</td>
</tr>
<tr>
<td>Listening 247 Threat Analyser</td>
<td>Amadeus, Atos, EDP</td>
</tr>
<tr>
<td>Threat Intelligence Integrator</td>
<td>Amadeus, Atos, EDP</td>
</tr>
<tr>
<td>Network-based Anomaly Detector</td>
<td>Atos</td>
</tr>
<tr>
<td>Skeptic II</td>
<td>Amadeus</td>
</tr>
<tr>
<td>User Behaviour Analytics Platform</td>
<td>Amadeus</td>
</tr>
<tr>
<td>SLiCER Cloud Archiver</td>
<td>Atos, EDP</td>
</tr>
</tbody>
</table>

Table 8 – Component validation test environments

On top of the comprehensive description of component interfaces included in this deliverable, the validation outputs will emphasize the methods used to guarantee the adaptability of those interfaces.
4 Integration Plan

This chapter defines the required interfaces (input and output) for component integration. The components presented in Chapter 2, initially developed and implemented in an isolated manner, will be integrated with the SIEM environments described in Chapter 3. Additionally, the interfaces between components will also be validated, making sure the operational context is valid and seamlessly working as a whole.

We will present the integration interfaces as described and represented in the DiSIEM architecture. It is relevant to note that not all components have direct integration with the SIEM, as in the case of the User Behaviour Analytics Platform.

4.1 Integration with SIEM

Component integration with the SIEM system is one of the core activities for the validation phase of the project. This integration can be accomplished directly between the developed component and the SIEM or in an indirect manner, through another component. In this section we focus on direct integrations between components and the SIEM.

4.1.1 Multi-level Risk Manager

The Multi-level Risk Manager component has a bidirectional integration with the SIEM platforms. The DBImport module of this component will ingest asset, vulnerability and incident data from the SIEM. The designed method is to use the file export capabilities of the SIEM in order to have the developed module accessing a file share or sFTP area, processing it and importing the data to the component's database.

The results of the risk assessment can be exported to the SIEM and a DBExport sub-component was developed for that purpose, using the same file export/import method. However, we will also implement a more efficient and seamless integration taking advantage of available SIEM connectors, having the SIEM accessing the database directly and pulling the relevant information from the database to the SIEM, bypassing the need for a file repository.

4.1.2 Diversity and Forecasting Analytics Platform

The Diversity and Forecasting Analytics Platform takes event data extracted from the SIEM directly from the connectors, for instance using API calls, performs the analysis on its dashboard sub-component, calls its engine sub-component for forecasting and displays the results back to its dashboard.

4.1.3 Threat Intelligence Integrator

We'll now detail, for each SIEM platform used by the DiSIEM partners, the designed integration plans for the OSINT Data Fusion & Analysis components, bearing in mind that they follow the DiSIEM principles stated in [D22], especially the one affirming that SIEM platforms should not be modified due to our extensions and no additional or significant manual work should be required to operate them:
• **XL-SIEM (ATOS):** ATOS XL-SIEM, can export and import events through the RabbitMQ\(^{13}\) Message Model. For this purpose, a new MISP module will be added to the MISP instance of the Threat Intelligence Integrator, allowing him to communicate with this SIEM using this message broker. The module will be also in charge of handling the conversion between the different standards used by MISP and the XL-SIEM for representing cybersecurity events;

• **ArcSight (EDP):** EDP ArcSight is a SIEM widely used in industrial environments. Indeed, MISP comes out with a specific module developed for the integration with ArcSight itself, exchanging information using the CEF format, supported by this SIEM. Another option could be letting ArcSight interact directly with the Threat Intelligence Integrator, for pulling/pushing specific events from the MISP database, relying on the REST API that MISP itself provides. In this case, the python library PyMISP\(^{14}\) could be used for making the integration process easier;

• **Elasticsearch (AMADEUS):** AMADEUS is relying on a local MISP instance, for their internal usage. The easiest way for performing the integration would be to synchronize the instance of the Threat Intelligence Integrator with this one, to exchange information in an automated way, through the built-in information sharing capabilities of MISP. Alternatively, as for ArcSight, Elasticsearch could pull/push data directly from the MISP database, or through some custom import/export module added to the instance of the Threat Intelligence Integrator.

4.1.4 Diversity-enhanced Monitoring

Skeptic II is an Event Generator, run in a different platform, writing aggregated application user sessions to the SIEM. Skeptic II natively supports JSON format and is shipped with an Elastic Stack connector. Custom SIEM connectors can be added to the Skeptic output module.

4.1.5 SLiCER Cloud Archiver

SLiCER has a passive interaction with the SIEM platform, simply receiving forwarded events. To integrate with SLiCER, it is necessary to configure the SIEM to forward security events to the defined IP address and network port, where SLiCER will be listening. The configuration may be changed in the “Slicer.conf” configuration file, either directly in the SLiCER folder or through its user Interface.

4.2 Component Integration

Some of the developed components were designed to be tightly integrated, continuously exchanging information in order to enable improved SOC operations. In this section, we describe how these component ensembles can be implemented and validated. Figure 12 represents the DiSIEM components, accentuating the components with active interfaces and numbering those interfaces.

---

\(^{13}\) [https://www.rabbitmq.com/](https://www.rabbitmq.com/)

4.2.1 Diversity and Forecasting Analytics Platform

The Diversity and Forecasting Analytics Dashboard retrieves the data from the SIEM and calls the Engine for the predictions, using interface 1. This functionality is implemented using a REST API. The modelling and analysis functionalities, that are part of the Engine, are called from a prototype Python Flask web service, offering one main endpoint, taking input data as a CSV or JSON and returning results as JSON. The modelling code in the Engine can also be run as self-contained modules and can stream results out in syslog format.

4.2.2 OSINT Data Fusion and Analysis Components

The integration with OSINT data fusion and analysis tools will rely on MISP. Both the Listening247 Threat Analyser and the OSINT Threat Detector can provide structured IoC, represented using the MISP JSON format, to the Threat Intelligence Integrator, associated to the OSINT sources they are using. These interfaces are identified as number 2 and 3, respectively. The integration plans will consider the following guidelines:

- **Listening 247 Threat Analyzer (DigitalMR):** the Listening247 Threat Analyzer is not using MISP. It is necessary to create a user, with limited privileges, in the MISP Instance of the Threat Intelligence Integrator. In this way, an authentication key has been generated, allowing this user to
directly interacting with the MISP Database, for pulling/pulling events remotely, helped by the PyMISP\textsuperscript{15} library;

- **OSINT Threat Detector (FCiências.ID):** the OSINT Threat Detector will also rely on a private MISP Instance. This simplifies the sharing process with the Threat Intelligence Integrator, which can be performed simply by synchronizing the two MISP instances, as explained in the MISP guide.\textsuperscript{16}

Moreover, for improving the representation of the shared IoC sets, specific custom MISP taxonomies will be created and integrated in the MISP instance of the Threat Intelligence Integrator. This will also help the analysis made by the latter, during the threat score computation.

### 4.2.3 Skeptic II

Figure 13 details the communication phases between the two components. The User Behaviour Analysis components ingest data from the SIEM extension Skeptic. The output of these components is displayed via the VASABI visualisation extension.

![Diagram of Skeptic II integration with User Behaviour Analytics Platform](image)

**Figure 13 Skeptic II integration with User Behaviour Analytics Platform**

\textsuperscript{15} [https://media.readthedocs.org/pdf/pymisp/master/pymisp.pdf](https://media.readthedocs.org/pdf/pymisp/master/pymisp.pdf)

5 Evaluation

In order to accurately validate the developed components and the overall success of the DiSIEM project, it is vital to define criteria that allows the consortium to measure the effectiveness of the planned activities, before they are started. This section defines some preliminary success criteria for component evaluation, aligned with the DiSIEM principles and methodology described in Section 1.1. The evaluation itself will be performed during the integration tasks and reported in deliverable D7.3 (validation results), on M36.

As described in Chapter 3, the component integration will use the available SIEM systems, being firstly implemented in the test environments. In order to be deployed in the production environment, the component must satisfy the success criteria defined in Section 5.1. This quality assurance mechanism is essential to guarantee that the critical production environments of industrial partners are in no way negatively impacted by possible bugs in the components, as well as making sure that the components have proven to be relevant for the SOC operations and that they do not harm the SIEM performance in any way.

5.1 Success Criteria

The purpose of each component was detailed in Chapter 2, presenting the motivation for the development and integration of the components with the SIEM platforms. First and foremost, the validation phase will determine if the purpose of the component was fulfilled and complied with the expected outcomes. This functional analysis is obviously the most relevant criterion and will precede any other evaluation. The success criteria parameters are presented in Table 9.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Success Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Component capabilities</td>
<td>The component operates as expected and provides the necessary outputs, enabling improved SIEM capabilities</td>
</tr>
<tr>
<td>Functional</td>
<td>SOC operation improvements</td>
<td>The SIEM end-users must evaluate the benefits of integrating the component in the SIEM platform. Performance improvements should be measured whenever possible</td>
</tr>
<tr>
<td>Technical</td>
<td>Complexity of SIEM integration</td>
<td>The component must be integrated considering the DiSIEM principles of least impact. The SIEM platform managers must validate that the integration effort required is adequate, considering the benefits of using the component and the improvements for the SIEM platform</td>
</tr>
</tbody>
</table>
### Technical Component adaptability

One of the DiSIEM cornerstones is that all developments must be conducted in order to allow the components to be implemented in multiple SIEM platforms. The component developers must measure the percentage of the component that needs to be redesigned or adapted for a different SIEM platform, to determine the adaptability of the component.

### Functional Relevance of the development

While the teams operating the SIEM are able to give direct and immediate feedback regarding the components, the most relevant success criterion is to determine if the developments contributed to improve the organization cybersecurity capabilities, including for C-level management roles.

<table>
<thead>
<tr>
<th>Technical</th>
<th>Functional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component adaptability</td>
<td>Relevance of the development</td>
</tr>
<tr>
<td>One of the DiSIEM cornerstones is that all developments must be conducted in order to allow the components to be implemented in multiple SIEM platforms. The component developers must measure the percentage of the component that needs to be redesigned or adapted for a different SIEM platform, to determine the adaptability of the component.</td>
<td>While the teams operating the SIEM are able to give direct and immediate feedback regarding the components, the most relevant success criterion is to determine if the developments contributed to improve the organization cybersecurity capabilities, including for C-level management roles.</td>
</tr>
</tbody>
</table>

**Table 9 – Evaluation success criteria**

### 5.2 SBD Validation Scenarios

In this section, we will apply the Scenario Base Design methodology to define some validation scenarios for one of our components: Skeptic II. Other components validation will follow similar ideas.

The first two phases of the SBD approach (Requirements and Design) will be covered in this report. The last phase (Evaluation) will be covered in the last validation deliverable.

During the Analysis Phase, a requirement analysis is performed. The idea is to understand the problem at hand and identify areas to improve. Various methods are used to study the situation and the context. Typically, these methods include interview with stakeholders, studies of the current situation and its implemented solution and brainstorming for other solutions. The findings were gathered and analysed in order to compose the scenarios that reflect important features of the problem, the typical and critical tasks of the users, the tools they use and the organisational context.

Following the Validation Methodology described in Section 1.1, we envision two validation scenarios for the Skeptic II extension:

- **Scenario 1:** User Activity Discovery and Exploration
- **Scenario 2:** Suspicious User Activity Detection

**Scenario 1: User Activity Discovery and Exploration**

A security analyst from the SOC team is performing manual investigation in order to analyze potentially suspicious activity noticed by a customer. He is looking to first confirm the suspicious activity and then investigate its origin. The analyst is performing semi-automated investigation, consisting of extracting different application logs from various locations, joining them and reconstructing the full
user activity flow. To perform this task, the analyst has to access the log archive network shares, identify the relevant application log archives and then perform **searches and parsing using different tools**. After extracting the relevant events for the period provided by the customer, the analyst notices that he cannot confirm the user activity seen in the logs is suspicious, he needs to extract more logs to compare the user actions seen with his past actions, but also compare it with other user activities with similar profile.

The analyst is particularly interested in the login time and location patterns, frequent actions, and activity spikes, to check the user history.

The security analyst needs to process large amounts of logs, so the work soon becomes tedious and time consuming. He decides to use the Elastic Stack to index all relevant application events and **explore the results visually** using Kibana. With Kibana dashboards built, the analyst quickly discovers that the user connection time and origin changed for the last month, but he cannot confirm that the user actions are illegitimate. He quickly realizes that he needs more advanced visualization capabilities mainly to make sense of the user action flow.

He suspects the user is performing fraudulent actions, but he needs more information about the functional flow of the application. **He gets in touch with the team in charge of the application** to better understand the action flow.

Finally, he is able to confirm that the user activity is fraudulent and that the user was exploiting a loophole in the application design. He opens a ticket and a subsequent patch was applied to the application and the customer was notified.

Although, the issue was fixed for the application in question, no changes were made to the different discovery and investigation tools. The analyst knows that not having a feedback loop to improve the efficiency of the investigation tools from the results of the past investigation is sub-optimal. Table 10 summarizes the claims for Scenario 1:

<table>
<thead>
<tr>
<th>Claim</th>
<th>Pros (+) / Cons (-)</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim 1. Investigation</td>
<td>(+) analyst has knowledge about certain features and metrics that can indicate fraud</td>
<td>Analytics</td>
</tr>
<tr>
<td>Approach</td>
<td>ulent activities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(+) the process to detect suspicious actions is semi-automated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-) the process is time-consuming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-) no possibility to run a detection approach for multiple application in a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>configurable manner</td>
<td></td>
</tr>
<tr>
<td>Claim 2. Visualizations</td>
<td>(+) the security analyst is able visualize application events with Kibana for</td>
<td>Visualizations</td>
</tr>
<tr>
<td>capabilities</td>
<td>better understanding of user activities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-) no advanced visualizations to reflect the functional flow of an application,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>peer/groups analysis.</td>
<td></td>
</tr>
</tbody>
</table>
**Table 10 – Scenario 1 Claims**

<table>
<thead>
<tr>
<th>Claim 3. Functional Knowledge</th>
<th>(-) no Kibana visualizations incorporated in centralized dashboards</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim 4. User history</td>
<td>(+) the analyst is highly dependent on functional knowledge; this slows down the investigation.</td>
<td>(-) partial user history</td>
</tr>
</tbody>
</table>

In the *Design Phase*, the project is moved from problem understanding to the envisioned solutions. Developers start by writing down the scenario in which they envision typical or critical services that people sought from the system and introduced concrete ideas about the new functionality.

For Scenario 1, the analyst would like to perform the same type of suspicious activity discovery and exploration in a more efficient way. When he receives notification for an alert for a specific application, he would like to have in place a tool that would allow him to have all relevant information for his investigation gathered in one place.

The analyst wants focus on the actual investigation and not on log extraction parsing or building visualizations. The tool would allow the analyst to select the application to investigate, the location of logs to ingest, the session features to extract and the time range he is interested in. Then the tools would gather the needed information and present it in a centralized visualization tool, such as Kibana, but with advanced visualization capabilities to allow the analyst to better understand the flow even if he is not familiar with the application itself.

The analyst wants the tool to allow a seamless integration of OSINT data feeds such as IP/Domain reputation, known vulnerabilities databases, targeted attack campaigns, etc.

The tool would allow the definition of custom rules to reflect expert knowledge helping the analyst understanding the user flow. Finally, the tool would have a feedback loop allowing the analyst to incorporate the knowledge gathered from previous investigations into the tools, this enriching its final output result.

**Scenario 2: Suspicious User Activity Detection**

In addition to the investigation and discovery tasks the security analyst performs on a regular basis, he receives alerts from some continuous user activity monitoring tools that he needs to check.

Most of the monitoring tools for suspicious activity detection are either rule based or use simple statistical models on basic application session metrics. Although having some basic monitoring in place improves the chances to detect and stop fraudulent user activities, the analyst knows that the detection rules and models are not frequently updated and cannot cover most of the potential attack scenarios.
Most monitoring tools use basic application log events and do not enrich the data ingested, therefore the analyst is still obliged to perform manual lookups from various OSINT sources to further inspect the alerts received.

Also, the user history is rarely taken into consideration by the monitoring tools, thus the analyst is still obliged to manually gather more data to build an approximate profile of the user activity patterns. Furthermore, alerts from the monitoring tools are usually presented in simple dashboards that lack advanced visualisations capabilities and proper reporting tools. The analyst is unable to display all the information he needs in a concise and friendly manner. Table 11 summarizes the claims from Scenario 2:

<table>
<thead>
<tr>
<th>Claim</th>
<th>Pros (+) / Cons (-)</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim 1. Reporting tools</td>
<td>(-) No proper reporting tool in place</td>
<td>Analytics</td>
</tr>
<tr>
<td>Claim 2. Visualisations capabilities</td>
<td>(+) Some visualization is possible in Kibana (-) the visualizations or dashboards do not contain all the needed information (-) the visualizations and dashboards are also not frequently updated (-) more advanced visualizations needed. These visualizations should contain concise information about the user session in a readable format.</td>
<td>Visualizations</td>
</tr>
<tr>
<td>Claim 3. Rules</td>
<td>(+) Some automated rules are in place (-) the rules are applied on simple features (-) no complex rules have been derived (-) significant amount of time to test the efficiency of a new rule (-) rules are not updated</td>
<td>Analytics</td>
</tr>
<tr>
<td>Claim 4. Detection Approach</td>
<td>(+) analyst has knowledge about certain features and metrics that can indicate fraudulent activities (+) the process to detect suspicious actions is automated (-) the process is time-consuming (-) UEBA can be applied to detect new suspicious activity patterns (-) no advanced features / metrics are used for the detection (-) many applications to monitor in parallel (-) need to have the possibility launch custom detection models/features on the application.</td>
<td>Analytics</td>
</tr>
<tr>
<td>Claim 5. User history</td>
<td>(-) user history is not considered</td>
<td>Analytics</td>
</tr>
</tbody>
</table>
Ideally, the continuous user activity monitoring tools would include both a rule-based and a behavioural based detection mechanism. The rule-based detection mechanism should allow for flexible rule management including frequent rule updates and fine-tuning.

The behavioural based mechanism should allow the definition of advanced UEBA models on custom session features and should be frequently updated and fine-tuned to reflect the current user action patterns.

The detection tools should also allow the analyst to have all the enriched results accessible from a centralized visualisation tool with advanced visualizations capabilities, allowing the SOC teams to have accurate reports and a fine-grained understanding of user activities.

The detection tools need to allow for an easy integration with external data sources such as OSINT data feeds, GeoIP lookup databases, etc.

Finally, an ideal user activity monitoring tool should be highly flexible, allowing the user to specify all parameters specific to a particular application monitored: input log location, UBA models, session features, UBA model parameters, Rules, session output location, score aggregation parameters, etc. Such scenarios and claims will be re-evaluated in the next deliverable, and the progress will be noted and measured.

### 5.3 Ethical and Legal Aspects

SIEM platforms are used by organizations to collect, process and store large quantities of security events, which include multiple types of information from numerous sources. While most of the collected information can be classified as technical data, the volume of private information, more specifically personal information, is also relevant.

The components developed in DiSIEM will collect and process data that is legally protected, for instance by the GDPR.\(^{17}\) Part of the information is originated in the organization’s internal systems (e.g., user activity, Internet access logs, physical access logs) but there is also information being collected from public networks, mainly the Internet (e.g., Twitter data, threat intelligence data). Regardless of the information origin, the developers must take into consideration privacy requirements when designing and implementing the components.

---

During the component integration and evaluation phase, ethical and legal issues may arise. While each partner organization is able to address legal questions to internal legal departments, it is our objective to maintain permanent contact with the Ethics advisor designated for the project. We will address any doubts in this field to the advisor, making sure that component implementation will not lead to any doubtful decisions regarding information processing and dissemination.
6 Conclusions and Future Work

In this report we summarized the DiSIEM previously defined architecture and detailed the role of each developed component, stressing their most relevant characteristics, objectives and interfaces. The validation environments made available by the three industry partners (Amadeus, Atos and EDP) were also detailed, allowing for a clearer vision on how the component demonstrators will be evaluated. The success criteria for the integration and validation phases were laid out, defining the key aspects that must be considered when evaluating the readiness and relevance of the components.

We believe that by integrating the components in the partner’s operational SIEM environments, DiSIEM will provide significant insights on their relevance to SOC operations, making the transition from theoretical models to real-world implementations, precisely as defined in the project goals. This document will be used as a reference for the integration activities of DiSIEM throughout the rest of the project.

Considering that the components will be integrated in the corporate environments of industrial partners, there are some inevitable limitations regarding the dissemination of results. The SIEM platform is, by definition, responsible for processing and storing critical information, including details for security incidents and vulnerabilities. Some of the components will explore SIEM data and predictably access confidential information. Therefore, when reporting on component evaluation, it is necessary to redact part of the information, without compromising the ability to assess the project results.

One of the consortium’s aspirations is to include relevant stakeholders in the component implementation and integration, maximizing the validation scenarios and welcoming useful feedback to improve the components. Each industrial partner will address this objective by interacting with both internal and external stakeholders, which may include clients, different business areas and partners. The most straightforward examples would be to include SOC clients, internal or external, and SIEM maintenance teams, especially if they include external or subcontracted resources. Results dissemination will follow the methodology defined for the DiSIEM project, as described in WP8 deliverables.
References


