D2.2
Reference Architecture and Integration Plan

The DiSIEM project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 700692.
Editor
Alysson Bessani, FCiências.ID

Contributors
Alysson Bessani, FCiências.ID
Fernando Alves, FCiências.ID
Adriano Serckumecka, FCiências.ID
Susana González Zarzosa, ATOS
Gustavo González Granadillo, ATOS
Ilir Gashi, City
Frances Buontempo, City
Cagatay Turkay, City
Phong Nguyen, City
Olivier Thonnard, Amadeus
Miruna-Mihaela Mironescu, Amadeus
Zayani Dabbabi, Amadeus
Abdullahi Adamu, DigitalMR
Michael Kamp, 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>FCiências.ID and all other partners</td>
<td>First version of the document, submitted to the EC</td>
</tr>
<tr>
<td>2.0</td>
<td>FCiências.ID</td>
<td>Following the recommendations of the project reviewers, we improved the reference architecture (Figure 2), its discussion, and did minor improvements on Figure 1 (project overview).</td>
</tr>
</tbody>
</table>
Executive Summary

The DiSIEM project aims to address several limitations of existing SIEMs already deployed in production by defining, designing, and implementing a set of extensions to be deployed together with these systems. This deliverable defines the reference architecture adopted in the project, discusses all the components (also called extensions) that group the technical innovations under development in the project, and present a preliminary version of the integration plan of these components.

The reference architecture (see figure below) takes into consideration the limitations of existing SIEMs, as identified in Deliverable 2.1. This architecture is based on the design of four principles – (1) no modification on existing SIEMs, (2) the existence of extension ports in SIEMs, (3) the existence of non-integrated components, and (4) no substantial additional configuration should be needed - which constraint the expected behaviour of the extensions under development in the project. The architecture considers nine components that encapsulate the technical contributions being developed in work packages 3-6, namely: Enhanced Application Monitoring, Network-based Behavior Anomaly Detector, Listening247 Threat Predictor, OSINT Threat Analyser, Context-aware Intelligence Integrator, Action Sequence Analysis for User Behavior Understanding, Visual Analysis of Diverse SIEM Data, Diversity Assessment and Forecasting, and Cloud-backed Long-term Events Archive. All these components can be integrated to SIEMs following three different patterns - Event Generator, Event Inspector, and Event Collector – that are supported by the four SIEMs selected for validating the project innovations (HPE ArcSight, XL-SIEM, Splunk, and Elastic Stack).
The deliverable also contains a preliminary version of the integration plan for the components. Within these plan, we assess the requirements for the components execution environments and discuss how they can be integrated with the SIEMs being considered in the project.

This architecture and integration plan, together with the analysis of the state of the art of the SIEM technology presented in Deliverable 2.1, correspond to the main result of the DiSIEM WP2 ("Requirements and Architecture for SIEM Integration"). The information here presented will guide the design and implementation of the technical innovations developed in the next two years of the project.
# Table of Contents

1  Introduction.................................................................................................................. 8
   1.1  Limitations of SIEMs............................................................................................. 8
   1.2  DiSIEM Reference Architecture and Components.............................................. 9
   1.3  SIEM Integration..................................................................................................... 9
   1.4  Organization of the Document............................................................................... 10

2  DiSIEM Reference Architecture.................................................................................... 11
   2.1  Main Objectives ..................................................................................................... 11
   2.2  DiSIEM Architecture............................................................................................. 12
   2.3  SIEM Integration Patterns...................................................................................... 14
   2.4  DiSIEM Components Overview............................................................................. 16
   2.5  Final Remarks ........................................................................................................ 18

3  DiSIEM Components .................................................................................................... 19
   3.1  Diversity-enhanced Monitoring............................................................................ 19
      3.1.1  Enhanced Application Monitoring................................................................. 19
      3.1.2  Network-based Behaviour Anomaly Detector ............................................... 21
   3.2  OSINT Data Fusion and Analysis .......................................................................... 22
      3.2.1  Listening247 Threat Predictor ....................................................................... 22
      3.2.2  OSINT Threat Analyser ................................................................................ 24
      3.2.3  Context-aware Intelligence Integrator .......................................................... 25
   3.3  Visualisation and Analysis Tools ......................................................................... 27
      3.3.1  Action Sequence Analysis for User Behaviour Understanding ..................... 27
      3.3.2  Visual Analysis of Diverse SIEM Data.............................................................. 28
      3.3.3  Diversity Assessment and Forecasting ............................................................ 29
   3.4  Infrastructure Enhancements ............................................................................... 31
      3.4.1  Cloud-backed Long-term Events Archive ..................................................... 31
   3.5  Final Remarks ........................................................................................................ 32

4  Preliminary Integration Plan.......................................................................................... 33
   4.1  Integration between Components ......................................................................... 33
      4.1.1  OSINT Data Fusion and Analysis .................................................................. 33
      4.1.2  Application Monitoring and User Behavior Analysis .................................... 33
      4.1.3  Diversity Analysis and Visualisation ............................................................... 34
   4.2  Integration with Selected SIEMs ......................................................................... 34
      4.2.1  Enhanced Application Monitoring ................................................................. 34
      4.2.2  Network-based Behavior Anomaly Detector ................................................... 34
      4.2.3  Listening247 Threat Predictor ...................................................................... 35
      4.2.4  OSINT Threat Analyser ................................................................................ 35
      4.2.5  Context-aware Intelligence Integrator ............................................................ 36
      4.2.6  Action Sequence Analysis for User Behaviour Understanding ..................... 36
      4.2.7  Visual Analysis of Diverse SIEM Data.............................................................. 37
      4.2.8  Diversity Assessment and Forecasting ............................................................ 37
      4.2.9  Cloud-backed Long-term Events Archive ..................................................... 38
   4.3  Final Remarks ........................................................................................................ 38

5  Summary and Conclusions ......................................................................................... 39

References ....................................................................................................................... 40
List of Figures

Figure 1 - DiSIEM Architecture...........................................................................................................13
Figure 2 - DiSIEM architecture and components..................................................................................17
Figure 3 - Enhanced Application Monitoring extension architecture..................................................20
Figure 4 - Network-based behaviour anomaly detector architecture.....................................................22
Figure 5 – Listening247 OSINT threat predictor proposal........................................................................23
Figure 6 – OSINT Threat Analyser architecture of our framework.......................................................25
Figure 7 - Context-aware Intelligence Integrator architecture.................................................................26
Figure 8 - Conceptual diagram of the component....................................................................................28
Figure 9 – Visual Analysis of Diverse SIEM Data.....................................................................................29
Figure 10 - The diversity assessment and forecasting tool for alert filtering, labelling and analysis ......................30
Figure 11 - Long-term Cloud Event Archival architecture.......................................................................32
List of Tables

Table 1 - Integration Patterns and the four SIEMs considered in DiSIEM.............16
Table 2 – DiSIEM components and their integration patterns........................................18
1 Introduction

Organizations currently monitor and manage the security of their infrastructures by setting up Security Operation Centres (SOC) to make security-related decisions (e.g., which system is under attack, what has been compromised, where has an access breach occurred, how many attacks have happened in the last 12 hours). A SOC obtains an integrated view of the monitored infrastructure by employing a Security Information and Event Management (SIEM) system. These are complex systems that incorporate the functionality to collect logs and events from multiple sources, correlate these events together and then produce summarised measurements, data trends and different types of visualisations to help system administrators and other security professionals. Due to the nature of the functionality of these systems (the number of systems that feed events to them, the different types of events they need to correlate etc.) they are complex and costly to deploy and maintain.

The SIEM market is a growing one. According to a recent Gartner report [Gartner 2016], in 2015 the SIEM market grew from $1.67 billion to approximately $1.73 billion. There are many high-quality products from large IT vendors. Examples are IBM QRadar,1 HPE ArcSight,2 Splunk,3 LogRhythm4 and AlienVault OSSIM.5 Overall, the Gartner report identified two main drivers for such growth: threat management and compliance. The spectrum of new attacks (with hundreds of novel kinds of malware each month, including the ones related with advanced persistent threats) and the complexity of the IT infrastructures require a well-structured and integrated monitoring of security events. Additionally, many industries have strict requirements for compliance [Swift 2010], especially when it comes to log management, which often mandates the need for integrated log management functionality, as provided by SIEM systems.

1.1 Limitations of SIEMs

Despite their widespread use and the impressive market growth, current SIEMs still have many limitations (see a more complete analysis in D2.1 [D21]):

1. The threat intelligence capability of SIEMs is still in its infancy. Consequently, the systems are unable to automatically recognize novel threats that may affect (whole, or parts of) the monitored infrastructure, requiring considerable human intervention to adapt and react to changes in the threat landscape. This happens despite the availability of rich and up-to-date security-related information sources on the Internet (e.g., social media, blogs, security newsfeeds), which current SIEMs are unable to use.

3 http://www.splunk.com/
4 https://www.logrhythm.com/
5 https://www.alienvault.com/products/ossim
2. **Current systems can show any “low-level” data related with the received events, but they have little “intelligence” to process this data and extract high-level information.** These low-level data (e.g., number of failed logins in a server) are only accessible and meaningful to a limited subgroup of system admins and are difficult to translate to high-level metrics for senior, C-level managers (such as executives and decision-makers who need to make decisions on security expenditure but may not necessarily be well versed in the technical details). This impact, for instance, the capacity of SOC coordinators to justify the return on investment in security for an organization.

3. **Most data visualisation techniques in current SIEMs are rudimentary.** Advanced data visualisation in current SIEMs is still limited. This can seriously impact the ability of the SOCs to deal with incidents as and when they happen, in a timely manner.

4. **The event correlation capabilities of SIEMs are as good as the quality of the events fed to it.** Imprecise events and alarms generated by imperfect monitoring devices will be taken as correct by the SIEM and the uncertainties associated with these events are never communicated.

5. **Due to storage and event processing constraints, SIEMs are incapable of retaining the collected events for a long duration.** This limits their use in conducting forensic investigations in the long run, for example on advanced persistent threats, or other historical incidents.

The DiSIEM project aims to address these limitations, by providing a set of tools and methods to enhance the technology available today. However, instead of proposing novel SIEMs or extensive modifications to existing ones, the project approach consists in extending current systems, leveraging their built-in capacity for extension and customisation.

### 1.2 DiSIEM Reference Architecture and Components

This deliverable describes the reference architecture of DiSIEM, together with the main components under development in the project. These components aim to improve the state-of-the-art in SIEM technology, solving or relieving the limitations just described, without requiring radical changes in existing SIEM deployments. Therefore, the project put a strong emphasis in clearly positioning these components with respect to the extensibility technology available on SIEMs.

### 1.3 SIEM Integration

In a previous deliverable for this work package [D21], we provided an analysis of seven prominent SIEM products that members of the project have some access, namely: HPE ArcSight, IBM QRadar, Intel ESM, Alienvault OSSIM, XL-SIEM, Splunk, and Elastic Stack.\(^6\)

---

\(^6\) Technically, neither Splunk nor Elastic Stack are complete SIEM products. However, since both platforms are being extensively used for implementing cybersecurity-related big data analysis (also, by some partners) we decided to include them in our analysis. During this report, we call them SIEMs (together with other products) just to simplify the exposition.
After careful consideration about the available infrastructures for pilot deployment that could be provided by the partners leading the technology validation work package (WP7), and the limited resources of the project, the consortium decided to focus on the design of components targeting immediate integration with four SIEMs: HPE Arcsight (EDP), XL-SIEM (ATOS), Splunk, and Elastic Stack (Amadeus). Although most of our work in the integration plan focuses on technological feasibility of the integration on these SIEMs, the reference architecture assumes only basic extensibility features are available on every SIEM we are aware of. The objective is to ensure the components might be integrated with other SIEMs if the need arises (e.g., for further exploitation).

1.4 Organization of the Document

Chapter 2 gives an overview of the main objectives of DiSIEM and describes how these objectives are translated to a reference architecture and a set of concrete components that can be integrated in existing SIEMs (and how this can be done). Chapter 3 describes – in high level – the nine components devised in the DiSIEM project, while Chapter 4 presents the integration plan for these components considering the four SIEMs selected for validating the project contributions. Chapter 5 presents the conclusions and outline future work.
2 DiSIEM Reference Architecture

SIEMs systems are fundamental for large organizations to implement integrated security monitoring and management policies. In this way, they represent the fundamental tool to which SOC analysts work on.

SIEMs were initially adopted by most companies for achieving compliance with some industry specific regulation [Swift 2010]. This was especially important to ensure all events collected by the multiple sensors deployed in the organization infrastructure are kept on a normalized format, in a centralized database, and can be presented in human-readable reports.

More recently, the increasing sophistication of threats against the organizations, the complexity of modern IT infrastructures, and the amount of cybersecurity-related information available on the web, has lead organizations to employ SIEMs also to correlate and integrate information coming from multiple internal and external sources. The objective is to detect and react, as fast as possible, to any threat or attack against the managed infrastructure.

The DiSIEM project recognize the complex environment in which SIEMs operate, and aims to improve several aspects of existing systems by providing a set of components that can be integrated to existing, already deployed, systems. In this context, the DiSIEM reference architecture aims to provide a blueprint for defining which components would be integrated, in which SIEMs, and how this should be done.

This chapter defines the main ideas for the project reference architecture. We start by a description of the architecture main objectives (Section 2.1), then describe the main principles and ideas behind it (Section 2.2), and some patterns and techniques for integrating components to a selected set of SIEMs (Section 2.3). We conclude the chapter with an overview of the components considered in the DiSIEM reference architecture (Section 2.4) and some final remarks (Section 2.5).

2.1 Main Objectives

One of the core ideas of the DiSIEM project is to enhance existing SIEM systems with several advanced capabilities, that will be encapsulated as components to be integrated in the system. These components can be grouped in five main advances beyond the state of the art:

1. **Integrate diverse OSINT (Open Source Intelligence) data sources** available on the web, such as the NIST’s National Vulnerability Database,\(^7\) vulnerability and patch databases offered by vendors; threat intelligence data that organisations share with each other (e.g., Internet addresses,

\(^7\) [https://nvd.nist.gov/](https://nvd.nist.gov/)
URLs and file reputation databases like SANS ISC, malware domains lists; security blogs and data streams from social networks (e.g., Twitter, LinkedIn), collaborative platforms used in the Dark Web (e.g., Pastebin), search engines and online repositories (e.g., Google Hacking Database, Shodan, RIPE/ARIN), standards-based Indicators of Compromise (IoC) (e.g., STIX and OpenIoC), and many others. This data needs to be fetched, analysed, normalised and integrated to identify relationships, trends and anomalies, and hence helping to react to new vulnerabilities in the infrastructure or even predict possible emerging threats against the monitored infrastructure.

2. Develop novel probabilistic security models and risk-based metrics to help security analysts to decide which infrastructure configurations offer better security guarantees and increase the capacity of SOCs to communicate the status of the organisation to C-level managers.

3. Design novel visualisation methods to present the diverse data sets, to better support the decision-making process by enabling the extraction of high-level security insight from the data which will be used by the security analysts operating the SIEM.

4. To increase the value of the events fed to the system we will integrate diverse, redundant and enhanced monitoring capabilities to the SIEM ecosystem. The idea is to have enhanced sensors and protection tools built using a set of diverse tools. Implementing these mechanisms requires probabilistic modelling of diversity for security to define which combinations of tools are more effective and how much improvement can be expected. Likewise, we propose to deploy and integrate novel application-based anomaly detectors and thus improve the SIEM’s visibility into the functional security status of these monitored applications and its users.

5. Add support for long term archival of events in public cloud storage services. To satisfy the security requirements of such data (which contains sensitive information), the events will be encrypted and distributed through multiple cloud providers (e.g., Amazon, Windows Azure, Google), employing techniques such as secret sharing and information dispersal.

To support these improvements on existing SIEM systems, we need a flexible architecture that is general enough to be employed for multiple components in different SIEMs. This architecture is presented in the next section.

2.2 DiSIEM Architecture

The advances proposed in the project will be materialized through a set of tools and components, in the form of plugins, that can be integrated into existing SIEM systems, as illustrated in Figure 1.

---

8 [https://isc.sans.edu/](https://isc.sans.edu/)
The figure shows all the improvements will be made around the basic SIEM functionality, without requiring modifications on the base system. This is the most fundamental architectural principle of DiSIEM:

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

This is feasible as the SIEMs and other big data platforms considered in the project (e.g., HPE ArcSight ESM, Splunk, IBM QRadar, OSSIM from AlienVault, Elastic Stack) all support extensibility. The components we will design will be integrated in these systems using their extension interfaces. More specifically, most SIEMs we are aware of support the integration of new connectors (e.g., HPE ArcSight Connectors [HPDC 2012], or the IBM QRadar Connector [IBM 2014]) to collect events and provide RESTful interfaces for accessing stored events [IBM 2014b].

The importance of this principle comes from the impact that SIEM-agnosticism can have on the exploitation of the project results. By supporting multiple SIEMs, we increase the possibilities of adoption of DiSIEM components by different organizations.

These extensibility mechanisms open the possibility of creating add-ons and extensions to existing systems. The DiSIEM architecture exploits this feature to enhance the quality of the events fed to the system through custom connectors, and provide new visualisation tools, collecting data from the SIEM data repository. Our second architectural principle defines how we plan to enhance the system.
This principle is important as it constraints the integration ports we are considering for SIEMs. We do not assume any kind of deep integration API that would allow the modification of event processing and storage for the system. On the contrary, this principle was designed by following the in-depth state-of-the-art study reported in DiSIEM Deliverable 2.1 [D21].

Even restricting the amount of extensibility, it is still possible that the integration of some of the improvements developed on DiSIEM will not be possible in some of the target SIEMs. However, these components need to respect some basic principles to be considered part of the DiSIEM ecosystem, as defined by the following principle:

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

The aim of this principle is to ensure devised components offer enhancements over the existing systems, complementing existing functionalities, and reusing information (e.g., configurations, assets, data sources) already provided for the base system.

Our last architectural principle rules out components that require extensive configuration or manual intervention during operation. In the age of big data and machine learning, the DiSIEM components should either exploit information already available in the SIEM or OSINT freely available on the web.

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

This principle is important as it forces the component developers to figure out ways to gather information from the network or the SIEM, without defining it as a configuration parameter of the system.

### 2.3 SIEM Integration Patterns

Having discussed the main architectural principles of the DiSIEM model, we are now equipped to describe how these principles can be applied to existing SIEMs. In this section, we discuss the main integration patterns for extending a subset of SIEMs with the components devised in the project.
Although based on the same conceptual framework, SIEMs are implemented in different ways, using different technologies. Nonetheless, all SIEMs with prominent position in the market support some form of extensibility. The components developed in DiSIEM aims to be integrated, individually or together, in most SIEMs, so we constraint the integration patterns to what is supported by most available systems.

Under these constraints, the three basic types of components that can be built for integration with existing SIEMs considered in the project are:

- **Event Generator**: The component is an event generator for the SIEM, i.e., the produced outputs from the system must be ingested by the SIEM as IoCs and can be visualised and correlated using standard SIEM capabilities. This is especially important for new intrusion detector systems or OSINT analysers, which feed SIEMs with new information;

- **Event Inspector**: The component needs to access the database associated with the SIEM to inspect its configuration data (e.g., assets and their properties) and the collected events. This is especially important for visualisation and analysis tools;

- **Event Collector**: The component needs to have access to all events (or at least all events for a given type) sent to the SIEM. This is important for side-systems that do additional processing of events, beyond what the SIEM is capable of.\(^9\)

We anticipate that some SIEMs will not provide external APIs for implementing an Event Inspector (e.g., HPE ArcSight), neither considers the notion of connectors (e.g., Splunk) for implementing Event Collectors. However, in both cases the components can still be integrated, as long as the SIEMs support one of the two integration patterns. In the former case, by collecting all events of potential interest sent to the SIEM and storing them on a separated database to be consulted when required by the Event Inspector. In the latter case, the integration can be done by inspecting the SIEM database periodically looking for the events of interest collected in the last time interval.

It’s worth remarking that it might be possible to implement deeper integrations than the ones provided by these patterns. For instance, new data visualisation and analysis dashboards can be integrated in the visualisation panel of SIEMs, however, it is important to not assume that when developing such components, not every SIEM support extensibility at this level.

Table 1 shows how the SIEMs that will be used to validate the DiSIEM components can support these integration patterns.

---

\(^9\) Notice that making every sensor send the event to multiple sites is not a solution, as the configuration/management burden will be too high in large infrastructures.
Table 1 - Integration Patterns and the four SIEMs considered in DiSIEM.

<table>
<thead>
<tr>
<th>SIEM</th>
<th>Event Generator</th>
<th>Event Inspector</th>
<th>Event Collector</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPE ArcSight</td>
<td>Use of Flex connectors to parse custom formats or the generation of events in an already supported format (e.g., CEF, syslog)</td>
<td>The information stored in this SIEM can only be accessed through its own products. An external agent can’t access it</td>
<td>The connectors can be configured to send received events to the SIEM and other (multiple) destinations</td>
</tr>
<tr>
<td>XL-SIEM</td>
<td>Events can be generated and encapsulated in the syslog format to be processed by XL-SIEM Agents</td>
<td>Through a Storm DRPC server, XL-SIEM will provide an API for supporting diverse types of queries over the stored information</td>
<td>No event consolidator defined. However, XL-SIEM Agents can be configured to send events to multiple destinations</td>
</tr>
<tr>
<td>Splunk</td>
<td>Events can be generated and encapsulated in a well known format (e.g., syslog), that can be understood by the system. Splunk can ingest data in arbitrary formats by using the Splunk Data API</td>
<td>Splunk events can be consumed by other applications using the Splunk Search API or leveraging Splunk capability to forward data to 3rd party system via TCP sockets or Syslog</td>
<td>No standard way for doing that. Nonetheless, heavy forwarders can be used for collecting and consolidating events from multiple sensors (that run universal forwarders)</td>
</tr>
<tr>
<td>Elastic Stack</td>
<td>Logstash support most data formats used today, but can also be extended with new plugins to support custom formats</td>
<td>The Elastic Stack data store supports a REST API for implementing several types of queries over the stored information</td>
<td>Logstash can be configured and/or extended to send received information to multiple recipients</td>
</tr>
</tbody>
</table>

The table shows that it is possible to seamlessly implement most component integration patterns with the SIEM. The only exceptions are the Event Inspector in HPE ArcSight and the Event Collector in XL-SIEM and Splunk. In all these cases, it is possible to circumvent these limitations by using alternative APIs/mechanisms, as explained before.

2.4 DiSIEM Components Overview

Taking into consideration the integration patterns described in previous sections and the architectural principles defined in Section 2.2, we devised a set of components to enhance the existing SIEMs with respect to all the limitations discussed in the introduction of this report. Figure 2 presents an overview of the consolidated architecture envisioned for the enhanced SIEMs.

The figure illustrates a deployment with all components being used and, in some cases, working together in an integrated way. In practice, we expect that different SOCs will pick different subsets of components that fit their needs and priorities.
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), that improve the capacity of the SIEM to archive events for long time. 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. All the components present in this figure will be described in the next chapter.

The figure also shows a high-level overview of the data flow between components and the SIEM ecosystem (see legend on the top). As can be seen, several DiSIEM components receive **raw events** from the managed infrastructure or the internet (e.g., syslog events, firewall alarms, OSINT notifications, news about vulnerabilities) and generate **enhanced events**, that can be consumed either by the SIEM or other DiSIEM components. Other components consume large volumes of data from the SIEM (**bulk data transfers**) for providing novel visualisation/analysis dashboards or storing such data in the cloud.
Table 2 shows which integration pattern corresponds to each of the nine components presented in Figure 2. For each component, we mark Yes if the component needs the integration pattern, Possibly if the pattern can be used as an alternative, and No if the component does not require the pattern. As can be seen, all the components can be integrated to SIEMs in some way.

<table>
<thead>
<tr>
<th>DiSIEM Component</th>
<th>Event Generator</th>
<th>Event Inspector</th>
<th>Event Collector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhanced Application Monitoring</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Network-based Behaviour Anomaly Detector</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Listening247 Threat Predictor</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>OSINT Threat Analyser</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Context-aware Intelligence Integrator</td>
<td>Yes</td>
<td>Possibly</td>
<td>Yes</td>
</tr>
<tr>
<td>Action Sequence Analysis for User Behaviour Understanding</td>
<td>No</td>
<td>Yes</td>
<td>Possibly</td>
</tr>
<tr>
<td>Visual Analysis of Diverse SIEM Data</td>
<td>No</td>
<td>Yes</td>
<td>Possibly</td>
</tr>
<tr>
<td>Diversity Assessment and Forecasting</td>
<td>Yes</td>
<td>Possibly</td>
<td>Yes</td>
</tr>
<tr>
<td>Cloud-backed Long-term Event Archive</td>
<td>No</td>
<td>Possibly</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2 – DiSIEM components and their integration patterns.

2.5 Final Remarks

This chapter described the reference architecture of DiSIEM, with its underlying principles and main components. Our exposition focuses on showing how the components devised in the project can be integrated in SIEMs already in production. We also provide examples of how the integration patterns adopted in the project can be implemented in the four SIEMs selected in the project: HPE ArcSight, XL-SIEM, Splunk, and Elastic Stack.
3 DiSIEM Components

In this chapter, we present the components being developed in the DiSIEM project. The descriptions are necessarily brief, just to position the component within the overall DiSIEM reference architecture. Furthermore, it is important to remark that these components are based on ongoing work in DiSIEM WPs 3-6, so it is reasonable to expect that some of the designs presented here will be subject to updates in the next two years of the project. These updates will be reported in future deliverables and papers produced in WPs 3-7.

3.1 Diversity-enhanced Monitoring

These components comprise a set of advanced sensors built either for applications or on top of existing basic network data collector services. The objective of these components is to feed information-rich events to the SIEM for improving its capabilities for discovering threats in the monitored infrastructure. This addresses the 4th limitation of current SIEMs, as described in Section 1.1.

3.1.1 Enhanced Application Monitoring

The Enhanced Application Monitoring component is an application-based intrusion/anomaly detector that aims at enhancing application security using a User/entity Behaviour Analytics (UBA) approach. This component will leverage both supervised and unsupervised machine learning techniques to compute an anomaly score for application sessions. The anomaly score will then be used to take appropriate action (e.g., warn, block, suspend). The extension design is modular to support different applications and to allow better interaction with the SIEM and other DiSIEM components.

The architecture of the components is described in Figure 3, and is composed by the following modules:

- **A configuration module**: The extension will rely on a single configuration file that will define its expected behaviour and its interaction with the SIEM and the rest of the SIEM diversity-enhancement components. The configuration will define the application log input sources, the analytical models, the scoring scheme and the output location (e.g., SIEM, Application);
- **An input module**: This module is used to parse and ingest application logs in addition to OSINT data, being useful to improve the anomaly detection performance;
- **Aggregation/Correlation/Normalization module**: This module will use the parsed application logs coming from the input module to perform the needed record correlation and normalization and create application sessions.\(^\text{10}\)

\(^{10}\) Notice this is not a general event correlation engine, as used in SIEMs, but rather a module for correlating application records to build a user session.
- **Behavioural engine:** This module is used to build the statistical models as a learning phase and to evaluate the anomaly score of application sessions against the previously built models. The behavioural engine will first be an implementation of the Skeptic framework [D61], however, other supervised or unsupervised analytical models can be used;

- **Rule engine:** Alongside the behavioural engine, the Rule Engine will leverage functional and expert knowledge by applying user defined rules on the application sessions. The rules consumed by the Rule Engine can either be “Fuzzy” (Membership functions, Fuzzy Logic, etc.) or “Static” (Yes/No). The results from both the Rule and Behavioural engines are the output of the Enhanced Application Monitoring component;

- **Output module:** This module is responsible for forwarding the results from the Behavioural and Rule engines to the SIEM. Depending on the configuration, the output module can also send requests to the monitored application itself to take corrective actions against user sessions if needed (e.g., logging-out and/or blocking a user).

The Enhanced Application monitoring component will get feedback from the visualisation and analysis tool (described in Section 3.3.1) to improve the analytical models for anomaly detection by tweaking the configuration parameters. More details about the enhanced application monitoring component can be found in the Deliverable 6.1 [D61].
3.1.2 Network-based Behaviour Anomaly Detector

The Network-based Behaviour Anomaly Detector is a DiSIEM component that can be used to **monitor network traffic that takes place in the managed infrastructure together with logs generated by applications of interest to detect, in real time, anomalies that might represent attacks to the infrastructure.** A common example of such anomalies is the observation of a significant deviation of regular traffic patterns in a network segment or coming from a specific server.

This component works in two phases: (i) learning phase, and (ii) operating phase. During the learning phase, the sensor uses machine learning algorithms to build a model for the behaviour of the network traffic based on the normal usage of applications by the different users, whereas in the operating phase the sensor detects deviations or anomalous behaviours. The detection is based on comparing the actual traffic against the built model.

The sensor receives as input a dataset containing legitimate and malicious traffic with some of the following features:

- **Number or incoming/outgoing connections** from, to or between servers running the applications;
- **Size of the packets** sent/received;
- **Duration of the connections** established between servers or between clients and servers;
- **Source/destination IP addresses and ports** of the connections;
- **Information related to the application** relevant for modelling its behaviour (e.g., URL or function invoked by the user) recovered from the logs of the applications being monitored;

The proposed sensor uses one class support vector machines (One-Class SVM) [Corinna and Vapnik 1995] as an unsupervised algorithm that learns a decision function for novelty detection (i.e., classifying new data as similar or different to the training set). As this is a type of unsupervised learning process with no class labels, the method takes as input an array $X$ and detects the soft boundary of that set to classify new instances as belonging or not to that set. The sensor will consider, for instance, the number of connections between the application server and the different servers in the monitored infrastructure where the application is running, the connections and traffic between the ports used by the applications, etc.

After learning the normal behaviour of the application, the sensor performs evaluations of a valid dataset containing legitimate and malicious traffic to classify as abnormal every set of packets that falls outside the boundaries of the developed model. The output of this component is therefore an indication of whether the analysed traffic is normal or anomalous.

The architecture of this component is depicted Figure 4, and more details about it can be found in Deliverable 6.1 [D61].
3.2 OSINT Data Fusion and Analysis

One of the pillars of DiSIEM is the use of OSINT for enriching the information provided by SIEMs. This is done through two components for gathering and analysing OSINT from multiple sources,\(^{11}\) and a component that integrates the processed OSINT with internal events and prioritizes threats. These components, which will be fully described in Deliverable 4.1 [D41], aim to address the 1\(^{st}\) and 2\(^{nd}\) limitations of current SIEMs, as described in Section 1.1.

3.2.1 Listening247 Threat Predictor

The Listening247 threat predictor will gather OSINT data from various sources including Twitter, boards, news, forums, and Instagram, among others, using the DigitalMR listening247 platform.\(^{12}\) This information will then be

---
\(^{11}\) Listening247 Threat Predictor is based on the adaptation of an existing digital market research service for analysing cyber-security related OSINT and, therefore, is firmly based on existing ready-to-market tools. In accordance with the project Description of Action, in DiSIEM we are also investigating other recent machine learning techniques to extract threat information from OSINT data. These techniques were encapsulated in a second component, the OSINT Threat Analyser.

passed through a custom machine learning pipeline, which will apply noise filtering, natural language processing, and threat likelihood prediction, as shown in Figure 5.

The key inputs for the listening247 will be the keywords representing the target monitored infrastructure. This is the first step to noise filtering which involves forming queries that disambiguate keywords that might be homonyms to other words. For example, “Windows” (the operating system), will yield information about windows which are used in buildings, among others. Forming specialized queries for these relevant infrastructure as keywords such as for the Windows operating system will narrow down the vast amounts of OSINT data available on the internet, thereby making the amount of data to be processed more manageable.

To process all these information, the OSINT data needs to be aggregated with respect to time so that information occurring within a similar time interval get grouped together. This allows for time series analysis of the data from various sources. Information could be aggregated by week, making it more likely to contain data with similar information. More specifically, various sources of data have different velocities and aggregating data by week makes it more likely for information related to each other to be grouped together. By velocity, we are referring to the rate at which data is being produced. For example, when there is a breakdown of WhatsApp, social media sources like Twitter, Instagram, and
Facebook are usually the first ones to report the news. Followed by news agencies on their sites, and blog articles that follow up on the event.

After this pre-processing, the system filters out noise, specifically information found to not be useful for the infrastructures of interest. This noise filtering model will need to be trained on many OSINT data annotated with relevance by human curators with domain knowledge. Filtered OSINT data will then be allowed to pass through to the second model, which will use natural language processing to obtain meta-data such as the infrastructure and possibly the locations involved, which will be added to the data in STIX v2.0 (JSON). Prediction of threat likelihood will require annotating data with threat for supervised training. As a result, the STIX payload could include information of threat likelihood to the infrastructure, and possibly a prediction confidence to avoid false alarms. This will likely involve the use of recurrent neural networks such as Long Short Term Memory Neural Networks or Gated Recurrent Neural Networks [Alpaydin 2014], which remember information over time and uses that to influence their next prediction. This is essential for events that unfold over time.

Finally, the threat predictor will send over the information, either directly to the SIEM or to the Context-aware Intelligence Integrator component (Section 3.2.3), in the STIX 2.0 format, which follows the JSON syntax.

### 3.2.2 OSINT Threat Analyser

In a similar way to the previous component, the objective of the OSINT Threat Analyser is to **gather and analyse security-wise web data relevant to the infrastructure monitored by a SIEM**. We consider that the SOC operators have a limited time budget to get in touch with the latest news. Therefore, this component aims at maximizing the amount of information gathered, while minimizing the amount of time required to view it. To achieve this objective, we propose a processing pipeline composed of an OSINT information gatherer, an automatic method for selecting the relevant information, and a summarizing function. More specifically, we use an automated tool to gather tweets from security-relevant accounts, a supervised machine learning technique (Support Vector Machines, Multi-Layer Perceptron Neural Networks, Deep Learning Neural Network Architectures [Alpaydin 2014]) to select the relevant ones for the specified infrastructure, and a clustering method ($k$-means) to avoid presenting repeated and/or unnecessary information.

Figure 6 shows the architecture of our initial solution, which focuses on Twitter. The data collector retrieves tweets concerning a specific keyword set (ideally the infrastructure to be monitored, but other relevant inputs can be given). The tweets are filtered as they mentioned the keywords (infrastructure elements) or not. Then, these are pre-processed into a normalized form before passing a feature extraction step. The feature vectors are fed to a supervised classifier that classifies the tweets as security relevant or not. Finally, there is a clustering step that gathers similar tweets, so that repeated information is not shown.
Differently from the previous component, our framework requires a set of accounts from where to gather the tweets. The accounts should be focused on cybersecurity (e.g., @USCERT.gov, @SecurityWeek) or related with the monitored infrastructure (e.g., @Cisco, @WordPress) to reduce the amount of non-relevant tweets gathered. It is possible to update the data collector to add/remove OSINT sources, as well as changing the assets monitored.

Furthermore, the component requires a keyword set describing the monitored infrastructure. Ideally, the framework could receive directly from the SIEM the various elements monitored, but this possibility is still being studied.

Although most of the current work on this component comprises Twitter monitoring, the component can also be used to monitor IP blacklists available online, through a different set of modules. This component aims to reduce the large false positive rate observed by industrial partners using these lists by integrating the 50+ data feeds available on the internet, improving the accuracy of the detected IoC.

### 3.2.3 Context-aware Intelligence Integrator

The Context-aware Intelligence Integrator aims to integrate OSINT security information (possibly from previous components) with threat intelligence collected from the local infrastructure and assign a threat score for the generated indicators of compromise. Therefore, despite its name, this component not only integrate information from multiple OSINT data sources, but also enriches this data with information from the monitored infrastructure.

This component will use heuristics to complement the static information about the monitored infrastructure with dynamic and real-time threat intelligence data generated by the SIEMs from inside their monitored infrastructure. Some examples of dynamic information coming from the infrastructure that could be used in the evaluation are:

- **Vulnerabilities** detected in the infrastructure that could appear as target of some IoC reported from OSINT data sources;
- **Entities or threat actors** involved in a previously detected incident in an incoming IoC;
- **Tools or intrusion sets** detected as installed inside the infrastructure are used as source in some incoming IoC;

- **IPs or domains** included in patterns provided by incoming IoCs (e.g., blacklisted IPs, such as the ones collected by the OSINT Threat Analyser component), which have already been reported as the source of an incident detected in the infrastructure.

The key feature of this component is to provide the SIEMs with a better evaluation of the *priority and relevance* of incoming security information received from OSINT data sources. Furthermore, it aims to provide a more meaningful threat prediction by being aware of what is happening or has been already reported from inside the infrastructure. With this purpose, it will also identify, analyse, and validate which are the sources of information and the attributes that better allows performing this classification of incoming OSINT data to assign a priority to the security information received. This component will focus on threat intelligence data provided from OSINT data sources and generated by the infrastructures following the standard STIX 2.0 format. Security information coming from OSINT data sources will be provided by other DiSIEM OSINT Data Fusion and Analysis components (see previous sections).

A draft of the architecture of this component is presented in Figure 7.

![Figure 7 - Context-aware Intelligence Integrator architecture.](image)
A threat score for the security information received from OSINT data sources will be calculated by this component. The final output will be a new STIX object equal to the original received from the other OSINT data Fusion and Analysis components, but now enriched with the threat score and the data related to the infrastructure used in its generation, as well as a reference to the original STIX object. The resulting JSON event with this STIX object will be sent using syslog to be integrated by the SIEMs or to be used by other DiSIEM OSINT-based components to refine their machine learning algorithms.

### 3.3 Visualisation and Analysis Tools

The Visualisation and Analysis Tools group comprises a set of components to enrich the dashboard of existing SIEMs, addressing thus the 2nd and 3rd limitations of current SIEMs, as described in Section 1.1.

More details about the components here described can be found in Deliverable 5.1 [D51] and in the upcoming Deliverable 3.1.

#### 3.3.1 Action Sequence Analysis for User Behaviour Understanding

This component aims to provide a comprehensive understanding of **user behaviour** through the visual analysis of their action sequences. Understanding user behaviour from action logs poses many challenges. First, the raw actions contain little semantics but they represent multiple levels of higher semantic concepts such as user tasks and roles. The number of actions is high and the sequences contain noise. Also, multiple facets exist in the data such as time, semantics and users. To address these challenges, we propose a visual analytics approach that combines the strength of both automatic and visual analysis methods to gain knowledge from data. Automated analysis techniques are applied to derive or mine high-level features from the raw data. The results are then visualised in a set of linked views enabling operators to interactively explore and gain deep understanding. One feature provides an **overview of sessions** (separation of action sequences such as based on user login and logout) according to multiple criteria. Another feature allows in-depth investigation of a subset of sessions through multi-scalar summaries of **action types and their temporal distributions**. The component also supports **interactive comparisons** to better understand anomalies, overlaps, and differences between sessions or users. If the actions contain anomaly model scores, the scores and associated uncertainties from underlying models will also be visually explored.

Figure 8 illustrates the Enhanced Application Monitoring component (Section 3.1.1) and how it interacts with the remaining of the system. The component reads raw log data from the application, extracts sessions and computes anomaly scores. It saves the result to the SIEM and this component takes those sessions as the input. Two types of output are produced in this component. The first one is the understanding about user behaviour gained by operators while they interact with the visualisations and the feedback. The second one is the feedback to the Enhanced Application Monitoring component to improve the performance of its analytical models.
3.3.2 Visual Analysis of Diverse SIEM Data

A large organisation usually makes use of multiple detection or monitoring tools to protect their network infrastructure. The generated data from those tools (stored in a SIEM) provides a chance for exploration and assessment of the performance of different combination of tools. This component supports visual analysis of such data coming from diverse security configurations.

First, the component provides an overall understanding of the monitoring network through visual summaries of high numbers of network activities and events, highlighting those alerted by security tools. It allows data exploration based on different perspectives such as alerts and security tools. Through an interactive network topology graph, the user will be able to visualise not only the alarms generated but also the anomalies detected in the traffic as well as additional threat intelligence information received from OSINT data sources and relevant to the different nodes. Besides, the user will have the possibility of generating new IoC associated to a specific node or the monitored network.

The component also provides visual representation and interaction mechanism to define what-if scenarios to compare and evaluate alternative configurations of security tools. Besides classic graphical means such as ROC curve [Collins 2014], novel visual representations will be developed to facilitate performance assessment. The component will be designed to handle both static and dynamic data. More importantly, time-varied performance could help operators assess and predict future configurations more effectively. External data sources such as OSINT data could also be used to identify any correlation with alerts.

Figure 9 presents an overview of the component. The Diversity Assessment and Forecasting component (described in next section) reads raw log data from SIEM and labels them with additional fields providing information about the systems raising alerts. This component takes those alerts and events as input and produces interactive visual representations supporting the designed tasks. The Context-aware Intelligence Integrator component (Section 3.2.3) provides threat intelligence from OSINT data sources after data fusion and analysis and including a threat score based on IoC provided from the monitored infrastructure. The
Network-based Behaviour Anomaly Detector component (Section 3.1.2) analyses the network traffic to detect anomalies in the normal learnt behaviour of the monitored applications. This component takes as inputs the output of those components to provide additional information to the situational awareness offered to the user through an interactive visual network topology graph.

3.3.3 Diversity Assessment and Forecasting

The Diversity Assessment and Forecasting component aids data collection and labelling, allowing assessment of diversity, and provides forecasting and prediction. The managed infrastructure will send data from more than one detection or monitoring tool for the same network or system to their SIEM. This will allow comparison or filtering of the alerts or events from the given number, N, of tools. A high-level metric showing how many times all the systems have “fired”/alerted together will be provided. Furthermore, different filtering rules will be available, allowing an architecture to be chosen from the possible configurations: one out of N (1ooN), majority, a given number, k (1<k<N), out of N (kooN), and everything (NooN). This feature can be used to select the most appropriate architecture of the given setup.

The alerts and events from each tool will have fields in common to facilitate their comparison. So far, we have built a working prototype using the Elastic Stack for two IDSs, Suricata and SNORT, which runs on a saved packet capture (pcap) file. The output from both were then sent via Logstash to Elasticsearch. The data (or “documents” in Elasticsearch parlance) is enhanced with a field to indicate from which system the alert came from. Our prototype shows the number of times both have alerted together with the number of times just one system has alerted.

This data tells us nothing about potential problems that have been missed, so this tool allows any saved system or network traffic to be labelled (offline) into malicious and non-malicious. The labelling could be done by various network traffic identifiers (protocol, src IP, dst IP, src port, dst port, timestamp, etc), or rules could be defined so labelling is done without further intervention from the
users of the assessment and forecasting module (e.g., always label traffic from this IP address on this port as malicious if IDS A has labelled it as such). The labels are values 1 or 0, indicating whether the traffic was malicious or not. Alternatively, a maliciousness/suspiciousness rating could be given (e.g., 0 to 1, with 0 being not malicious with high certainty, and 1 being malicious with high certainty). Using a numeric value allows a rating or cost to be assigned the problematic traffic that can be used in the metric plugin, described later.

The Diversity Assessment and Forecasting component will use the data to produce various statistics related to risk. For example, an assessment of sensitivity and specificity [Alpaydin 2014] of single protection systems and diverse combinations of protection systems based on their False Positive/False Negative rates can be made for labelled data. Even without labelled inputs, statistics and visualisations using the tools described in Section 3.3.2, can be informative. The outputs will indicate uncertainty, for example of the accuracy of a 100N configuration of IDSs. In addition to assessment, the package will make forecasts and predictions of events of interest using data from the labelled traffic and statistics from the diversity assessment steps. It will report the time to the next expected false positive (an alert which can safely be ignored) or false negative (an alert has not been generated but something untoward is likely to happen). It will report the frequency of false positive and false negatives over the next time interval. Other metrics of interest such as accuracy and frequency of alerts for one or more systems, or similar, can also be reported. Furthermore, these results will aid a SOC in answering questions such as “given a choice of diverse configurations, and a set of criteria (e.g., minimise false positives, minimise false negatives), which diverse configurations should I use”? Finally, this feature will assess the predictive accuracy and address calibration of the models developed.

Figure 10 depicts the flow from the saved, labelled data to the SIEM front end, via a separate tool for analysis, labelling, forecasting and prediction.

![Figure 10 - The diversity assessment and forecasting tool for alert filtering, labelling and analysis.](image)
3.4 Infrastructure Enhancements

The last component group is currently composed by a single component, that integrates SIEMs event archival capabilities with cloud storage services. This addresses the 5th limitation of current SIEMs, as described in Section 1.1.

3.4.1 Cloud-backed Long-term Events Archive

The purpose of the Cloud-backed Long-term Event Archive component is to store the 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 and Azure Blob Storage to 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 will organize and store the events in a cloud-of-clouds [Bessani et al. 2013, Oliveira et al. 2016], ensuring security, cloud fault tolerance, and cost efficiency. The clouds used for storing the events should be compliant with the EU General Data Protection Regulation, or any other legislation applicable to the personal or sensitive data. In the end, the objective is to eliminate the need for SIEMs to have a local archival infrastructure.

Figure 11 illustrates how the proposed component interacts with existing SIEMs. In the figure, the Sensors (S) generate events and send them to Connectors, which normalize and forward them to SIEM and to a Local Archive, if available. One possibility is to transfer the archived events from the Local Archive to the cloud store after an initial storage period (Figure 1a). If the Local Archive is not available, the events can be sent directly from the Connectors to our component at the same time they are sent to the SIEM (Figure 1b). The Event Archiver receives the events, groups them during a time interval (e.g., one hour), and creates an event block that is written in the clouds using, for instance, a cloud-of-cloud system designed in a previous project [Bessani et al, 2014].

---

13 https://aws.amazon.com/s3/
14 https://azure.microsoft.com/en-us/services/storage/blobs/
15 http://www.eugdpr.org/
To minimize the costs of using cloud storage services, it is important that the component organizes the events in such a way that typical searches on the event archival avoid downloading high volumes of data. This requirement comes from the fact that download bandwidth is the most expensive resource on most cloud storage services. To do that, 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 index files for the blocks considering event properties normally used for performing searches (e.g., the IP address of the event source).

Archival searches are executed in the component by using a REST interface, provided by the Query Manager. Read event blocks are cached to be read locally, as archival searches are usually part of a group of many operations, usually during a forensics analysis.

More details about this component can be found in Deliverable 6.1 [D61].

### 3.5 Final Remarks

In this chapter, we presented an overview of the nine main components that integrate the innovations being developed in DiSIEM. The objective was to discuss the main problems these components solve and how they interact with the SIEM ecosystem. In the next chapter, we discuss the integration of these components with the four target SIEMs being used in the project.
4 Preliminary Integration Plan

This chapter defines and validates the required interfaces (input and output) for the selected SIEMs to support the integration of each of the nine components described in the previous section.

The “integration plan” discussed in this chapter focus on the technical aspects of the integration. Our main objective is to assess, as early as possible, the feasibility of the integration of the DiSIEM components with the SIEMs to be used in the validation scenarios. Other aspects of the integration tasks (to be started in month 22) will be described in Deliverable 7.1 (month 24). Furthermore, the risks associated with the integration are discussed in Deliverable 9.2 [D92].

In the following, we first describe the plans for integrating some components between themselves (Section 4.1), and later we proceed to describe how each of the components can be integrated with the four selected SIEM (Section 4.2).

4.1 Integration between Components

Certain components were designed to work together. In this section, we describe how these component ensembles can be built.

4.1.1 OSINT Data Fusion and Analysis

Context-aware Intelligence Integrator can receive relevant OSINT info from both DigitalMR OSINT Threat Predictor and OSINT Threat Analyser, as well as IoC provided from the monitored infrastructure, enrich them with a threat score and then send the information to the SIEM or to other visualization/analysis tool.

All this data will be exchanged using the STIX 2.0 format and a REST API provided by the Context-aware Intelligence Integrator to receive IoC, such as:

- Threat intelligence generated from inside the infrastructure (e.g., from anti-virus software or firewalls deployed throughout the network);
- OSINT data, possibly already processed by DiSIEM components like Listening247 Threat Predictor and OSINT Threat Analyser.

The enriched IoC generated by the Context-aware Intelligence Integrator will be then forwarded to the SIEMs, as described in Section 3.2.3.

In terms of integration with the visualization components, the STIX 2.0 data follows a JSON format, which can be understood by popular visualizations tools such as D3.js or Kibana. It could also be used with Logstash for further processing to summarize the data prior to visualization as well.

4.1.2 Application Monitoring and User Behavior Analysis

The Action Sequence Analysis for User Behavior Understanding component may take as input the output of the Enhanced Application Monitoring component: action sequences extracted into sessions. Each session contains meta
information such as the user and the machine’s IP address, and the main information about the time and type of each action performed in the session.

4.1.3 Diversity Analysis and Visualisation

This component can be used to visualize multiple types of data. For instance, it can receive diversity analysis such as assessment of true/false positive/negative rates and predictions from models produced by the Diversity Assessment and Forecast component. The rates could be generated by a stand-alone process generating a daily CSV file or sending JSON on demand. Similarly, assessments, forecasts, or predictions can be sent to file or as JSON via a REST API from the Diversity Assessment and Forecast component to the visualisation component.

Another possible data source for visualisation could be provided by the OSINT-related components, following the STIX 2.0 format.

4.2 Integration with Selected SIEMs

In the following we discuss the preliminary integration plans for each of the envisioned components being developed in the project, with respect to four SIEMs: HPE ArcSight, XL-SIEM, Splunk, and Elastic Stack. As can be seen by the provided description, our preliminary analysis indicates that all components can be integrated with all these SIEMs.

4.2.1 Enhanced Application Monitoring

This component enhances the SIEM capability for monitoring existing applications, providing means for User Behavior Analysis of these applications.

**Execution Environment.** This component is developed in Java using the Apache Spark platform. Therefore, the ML computations executed may be CPU/RAM intensive depending on the volume of logs generated by the monitored application.

**Integration with SIEMs.** This component is an Event Generator, it uses the syslog protocol [Lonvick 2001] to communicate the generated alarms/events to the SIEM. All SIEMs considered in DiSIEM can work with syslog. HPE ArcSight, Splunk, and Elastic Stack have connectors/components that natively interpret this protocol, namely, Smart Connector Syslog [HPE 2017], Syslog-ng [Splunk 2016], Logstash syslog module [Elastic 2017], respectively. For XL-SIEM, two solutions are available: (1) the events are sent to the rsyslog daemon, which is available for all UNIX-like OS, which writes them to a file that can be read and parsed by the XL-SIEM Agent; or encapsulate the events in JSON and send them to the RabbitMQ server of XL-SIEM.

4.2.2 Network-based Behavior Anomaly Detector

This component monitors the network and applies machine learning techniques to detect traffic anomalies that may indicate attacks or other problems in the application host.
**Execution Environment.** Real-time analysis of the network traffic against the application behavior models obtained after training. To implement this monitoring for high-speed networks it may be necessary to have a cluster of servers (e.g., [Viegas et al. 2017]).

**Integration with SIEMs.** This component is an *Event Generator*. The events generated by the component with the result of the analysis of the traffic will be sent to the SIEM using the syslog protocol, in the same way as described in previous section.

### 4.2.3 Listening247 Threat Predictor

The DigitalMR platform provides a concise summary of the OSINT information related with the cybersecurity of the monitored environment. This component is a subscription service that can be personalized for each client.

**Execution Environment.** For components or applications connecting to the threat predictor; the main requirement for execution will be the capability of making REST API calls and processing the resulting data. It requires support for processing the resulting JSON (i.e., in STIX format).

The OSINT will be stored and processed in the cloud. The processing will involve filtering noise, aggregating data, and making decisions about the nature of the present data (i.e., threat or no threat) and potentially the likelihood of threats based on previous data.

**Integration with SIEMs.** This component is also an *Event Generator* that can be used by following a subscription model. The IoC generated by this component will be sent to the Context-aware Intelligence Integrator, if available (see Section 4.1.1), or directly to the SIEM. In the latter case, the syslog protocol will be used, in the same way as described in Section 4.2.1.

### 4.2.4 OSINT Threat Analyser

This component is like the previous one, but incorporates several experimental techniques for processing different kinds of OSINT data sources.

**Execution Environment.** The execution environment consists in one or more servers running the Apache Spark framework and some data collectors for obtaining OSINT from registered data sources. If a large amount of OSINT is analysed, this(ese) server(s) needs to have a high processing and storage capabilities.

**Integration with SIEMs.** The OSINT Thread Analysed component is also an *Event Generator*. The integration can be done as in the previous component.
4.2.5 Context-aware Intelligence Integrator

The output STIX objects including the threat score calculated by this component will be sent to the SIEMs using the SYSLOG/CEF format. They can be also stored locally in a database to be accessed directly by the SIEMs or other visualization components.

Additionally, the component could provide an interface where the SIEMs or other DiSIEM components can subscribe to a message queue (e.g., RabbitMQ) to send/receive IoC if this would be more suitable for the DiSIEM validation environments.

Besides the direct integration with SIEMs, the IoC can also be retrieved directly from the component database by custom connectors or modules available in the SIEM system.

Execution Environment. This component requires a local storage service (e.g., a relational database) and decent processing capabilities to be able to correlate and attribute a threat score to each enriched IoC computed.

Integration with SIEMs. This component is both an Event Generator and an Event Collector. The enriched IoC generated by the component will be sent to the SIEM using the syslog protocol, in the same way as described in Section 4.2.1. For collecting events, selected sensors can be configured to send messages directly to the component or the methods for implementing Event Collectors or Event Inspectors in Table 1 can be used.

4.2.6 Action Sequence Analysis for User Behavior Understanding

This component comprises a visualization dashboard that provide a comprehensive understanding of user behavior through the analysis of their action sequences during a session (i.e., user actions between its login and logout).

Execution Environment. This component is implemented using web technology, such as JavaScript, CSS, and Java, by using libraries such as D3.js.

Integration with SIEMs. This component is an Event Inspector. Its integration with existing SIEMs should, in principle, be based on the dashboard level, which might require different measurements for different SIEMs.

- HPE ArcSight: Dashboards in ArcSight are implemented using Java, but this component will be implemented using web-based technologies (e.g., JavaScript, HTML, CSS). Therefore, it will not be integrated tightly as a dashboard view in ArcSight. Instead, it reads and visualises data from the same sources that ArcSight ingests data. In that way, ArcSight and the component work on the same dataset.
• **XL-SIEM**: Dashboard in XL-SIEM are implemented using web-based technologies like this component. Therefore, it will be possible to integrate it as a view in XL-SIEM dashboard. However, minor changes in the dashboard source code might be needed to enable that. For this SIEM, this is not a problem as ATOS controls the source code of the system.

• **Elastic Stack and Splunk**: This will be the most straightforward integration among the considered SIEMs because both systems use Kibana for visualisation purposes. Kibana supports custom plugins that can easily accommodate the visualisation component. The idea is to convert the component to a Kibana plugin, which may be practical since Kibana also uses D3.js as in the component.

If some of these integrations prove to be too difficult, we can implement the dashboard as an independent application that follows the Event Inspector integration pattern defined in Table 1.

4.2.7 Visual Analysis of Diverse SIEM Data

The data generated from sensors and sent to SIEMs provide a chance for exploration and assessment of the performance of different combination of tools. This component supports such exploratory visual analysis of configurations using production data.

**Execution Environment.** Like the previous component, this one is also an advanced dashboard, so the environment and technology employed is the same.

**Integration with SIEMs.** The provided dashboard can be integrated in the SIEMs dashboards by the same means described for the previous component.

4.2.8 Diversity Assessment and Forecasting

The Diversity Assessment and Forecasting component aids data collection and labelling, allowing assessment of diversity, and forecasting and prediction.

**Execution Environment.** This component requires a local database and an application server that runs a set of scripts for analysing the data, generating forecasts.

**Integration with SIEMs.** This component is both an Event Collector and an Event Generator. Each SIEM already collects events. The alert filtering, labelling and assessment stage will read the data using the collectors and inspectors provided. The forecasting and prediction feature can send data via syslog to the SIEM, as described in Section 4.2.6, and to the visualisation component (see previous section) using CSVs or JSON, described above.

• **HPE ArcSight**: This can be configured to read forecasts and predictions sent via syslog;
- **XL-SIEM**: Extra fields for the status of events will be added and labelled directly in the front end. The assessments, forecasts, and predictions will be sent to the visualisation component;

- **Elastic Stack and Splunk**: Logstash will be used to add data to Elastic Search, using the `update_by_query` API to enrich previous events with data from diverse sensors. CSV and JSON data can be sent to these SIEMS allowing uploading of labelled data and predictions provided as JSON could be visualized directly or sent to the visualisation component.

### 4.2.9 Cloud-backed Long-term Events Archive

This component can be used to store SIEM-collected events in the cloud in a secure, dependable, and cost-efficient manner. The objective is to archive these events for more time, enabling long-term forensic analysis.

**Execution Environment.** The component is being developed in Java, and requires a certain amount of local storage for caching data. The amount of required processing should be moderate as the data need to be compressed, encrypted, and encoded before being written to the cloud storage providers.

**Integration with SIEMS.** This component is an *Event Collector* (although it can also be deployed as an *Event Inspector*, as described in Figure 11). Therefore, the integration of the component with the SIEMs requires the reception of the events sent to the SIEM. This is done in different ways for different systems, as described in Table 1.

### 4.3 Final Remarks

This chapter presented an overview of the integration plan for the nine components currently devised for the DiSIEM project. This integration plan comprises both the integration of components that can work together and their integration with the four SIEMs employed for validation in the project: HPE ArcSight, XL-SIEM, Splunk, and Elastic Stack.
5 Summary and Conclusions

This deliverable defined the reference architecture adopted in the DiSIEM project, discussed all the components (also called extensions) under development in the project, and presented a preliminary version of the integration work plan of these components (between themselves and with a subset of selected SIEMs).

Building upon the main limitations of existing SIEMs (initially identified in Deliverable 2.1 [D21]), we defined four high-level architectural principles for SIEM extensions. These principles lead to a general reference architecture (Figure 2) in which the technical contributions being developed in work packages 3-6 are grouped in nine components. All these components can be integrated to SIEMs following three different patterns - Event Generator, Event Inspector, and Event Collector – that are supported by the four SIEMs selected in the project (Table 1).

The deliverable also contains a discussion about the technical feasibility of the integration of each component with existing SIEMs. Within this discussion, we assess the requirements for the components execution environments and discuss how they can be integrated with the SIEMs being considered in the project. These drafted plans are necessarily preliminary as most of the components are still in their early prototype stage, and the components integration tasks (T4.4, T5.4, and T6.4) start only on month 22. Therefore, all these plans will be re-evaluated and further refined at this point, when the components are expected to be ready for initial deployment. These integration activities will be reported in Deliverable 7.1 (“Validation Plan”), in the month 24 of the project. For a discussion about the risks associated with the integration, we refer the reader to Deliverable 9.2 [D92].

The results presented in this report, together with the analysis of the state of the art in SIEM technology presented in Deliverable 2.1 [D21], corresponds to the main results of the DiSIEM WP2 (“Requirements and Architecture for SIEM Integration”). These results will guide the design and implementation of the technical innovations developed in the next two years of the project.
References


