D5.3
Fully operating, integrated visualisation system with diverse SIEMs

Project Number 700692
Project Title DiSIEM – Diversity-enhancements for SIEMs
Programme H2020-DS-04-2015

Deliverable type Demonstrator
Dissemination level PU
Submission date August 31, 2019 (M36)

Responsible partner Fraunhofer IAIS
Editor Michael Kamp, Cagatay Turkay
Revision 2.0

The DiSIEM project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700692.
Editor
Michael Kamp, Fraunhofer
Cagatay Turkay, City

Contributors
Michael Kamp, Fraunhofer
Cagatay Turkay, City
Siming Chen, Fraunhofer
Linara Adilova, Fraunhofer
Ilir Gashi, City
Zayani Dabbabi, Amadeus

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Michael Kamp</td>
<td>First draft of the document</td>
</tr>
<tr>
<td>0.12</td>
<td>Linara Adilova</td>
<td>Adding technical details</td>
</tr>
<tr>
<td>0.13</td>
<td>Siming Chen</td>
<td>Adding technical details</td>
</tr>
<tr>
<td>0.2</td>
<td>Zayani Dabbabi</td>
<td>Draft of use-cases</td>
</tr>
<tr>
<td>0.3</td>
<td>Michael Kamp</td>
<td>Draft of executive summary</td>
</tr>
<tr>
<td>0.4</td>
<td>Linara Adilova</td>
<td>Edited the description of UBAP</td>
</tr>
<tr>
<td>0.5</td>
<td>Linara Adilova</td>
<td>Added how to install and run</td>
</tr>
<tr>
<td>0.6</td>
<td>Michael Kamp</td>
<td>Adding conclusion and summary</td>
</tr>
<tr>
<td>0.7</td>
<td>Phong Nguyen</td>
<td>Adding details for VASABI and Diversity Platform</td>
</tr>
<tr>
<td>0.8</td>
<td>Cagatay Turkay</td>
<td>Revising overall and adding details to UBAP</td>
</tr>
<tr>
<td>1.0</td>
<td>Cagatay Turkay</td>
<td>Revising in response to reviews</td>
</tr>
<tr>
<td>1.1</td>
<td>Linara Adilova</td>
<td>Revising in response to reviews</td>
</tr>
<tr>
<td>1.2</td>
<td>Siming Chen</td>
<td>Revising in response to reviews</td>
</tr>
<tr>
<td>2.0</td>
<td>Cagatay Turkay</td>
<td>Revisions merged</td>
</tr>
<tr>
<td>2.1</td>
<td>Cagatay Turkay</td>
<td>Further revisions made to the narrative</td>
</tr>
</tbody>
</table>
Executive Summary

This deliverable presents the final visualization systems, the scenarios in which they can be applied, and how they can be used in practice. Moreover, it provides a brief description of application scenarios in which the systems have been applied in practice (for a more detailed description the reader is referred to deliverable D7.3 [D73]). In WP5 two visualization systems/components were developed, the User Behaviour Analysis Platform and the Diversity and Forecasting Analytics Dashboard. Both systems analyse SIEM data, visualize it and apply machine learning techniques to allow obtaining a deeper understanding of events within the system. At the same time, they support security operators in decision making and model building, substantially facilitating these time-consuming tasks. The systems have been designed and adapted giving utmost priority to the requirements of SIEM users in the consortium, according to the feedback obtained from extensive testing of the systems by our SIEM user partners. This practical evaluation shows the significant impact these components have on the work of SIEM users and show that they have become an integral part in their SIEMs.

The visualization systems – published as open source\(^1\) – are general purpose components that can be applied in a wide range of application scenarios; a specification of suitable scenarios is provided for each component. While both systems have been adapted to the requirements of our SIEM user partners in the form of demonstrators (described in deliverable D7.3), the open-source variants contain all features and techniques developed in this work package. Moreover, the systems are built up of modular components and can be tailored to the needs of different SIEM users. Thus, the systems have a very high potential to have a substantial impact beyond the scope of the project. To support that, the partners committed to continue their work on maintaining and enhancing the systems.

\(^1\) https://github.com/disiemprojecteu
# Table of Contents

1  Introduction .................................................................................................................. 8  
2  Description of Visualization Systems ......................................................................... 9  
   2.1  User Behaviour Analysis Platform (UBAP) ............................................................ 9  
   2.2  Diversity and Forecasting Analytics Dashboard ....................................................... 15  
3  Integration of Visualization Systems ......................................................................... 18  
   3.1  UBAP ...................................................................................................................... 18  
      3.1.1  General Requirements .................................................................................... 18  
      3.1.2  How to Install and Run .................................................................................. 19  
      3.1.3  Integration with Partners ............................................................................... 26  
   3.2  Diversity and Forecasting Analytics Dashboard ....................................................... 26  
      3.2.1  Requirements ................................................................................................... 26  
      3.2.2  How to Setup ..................................................................................................... 26  
      3.2.3  Integration with Partners ............................................................................... 28  
4  Usage of Systems in Production ................................................................................. 29  
   4.1  Use-Cases for the User Behaviour Analysis Platform (UBAP) ................................. 29  
   4.2  Use-Cases for the Diversity and Forecasting Analytics Dashboard ......................... 30  
5  Summary and Conclusions ......................................................................................... 32  
References ....................................................................................................................... 33
List of Figures

Figure 1 - Illustration of the interaction of all four components (with U4 and VASABI as separate sub-components) of the user analysis platform with a common data storage. High resolution sub-figures of the components can be referred later, including Topic modelling-based user behaviour (Figure 2) and U4/VASABI (Figure 5 and 6).................................9
Figure 2 - Visual analytics system overview, including LDA modeling part (backend), Topic distribution view (a), topic-action matrix view (b) and chord diagram view (c). The action class is shown in (d).........................................................10
Figure 3 - An illustration for behaviour analysis with LDA topic modelling........12
Figure 4 - Workflow for demonstrating how to use the visual analytics tool for topic modelling-based user behaviour analysis (UBAP-1).......................12
Figure 5 - The interactive data visualization interface for U4 (UBAP-3) where the emphasis of analysis is largely on individual users and their sessions........13
Figure 6 - The interactive data visualization interface for VASABI (UBAP-4) where the emphasis of analysis is on the collective analysis of users through clusters of users (computed through a technique building on the results of UBAP-1) and understanding individual user behaviour.................................................14
Figure 7 - Diversity and Forecasting Analytics Dashboard showing the events on a temporal axis along with summary views to highlight the IP and domain range, as well as visualizations to indicate multiple modelling results........16
Figure 8 - Workflow for demonstrating how to use the visual analytics tool for topic modelling-based user behaviour analysis (UBAP-1)..................20
Figure 9 - The web-based LDA visualization working integrated with Amadeus's live data stream through a connection with the Elasticsearch index..............20
Figure 10 - The data flow setup to integrate U4 and VASABI interfaces on top of live elasticsearch stack as is the case for Amadeus’s setup.....................22
Figure 11 - The U4 interface running on live data drawn from the ElasticSearch index for Amadeus. In this screenshot, multiple users have been selected on the top Session Overview menu and the sessions from the selected users are visualized within the Session Timeline.................................................22
Figure 12 - The web-based U4 solution working integrated with Amadeus’s live data stream through a connection with the Elasticsearch index..............24
Figure 13 - First step is to select a time window for further investigation and the User Overview is populated with user data where each bar represents a single user coloured with the average score of their sessions. In this view, the users are grouped according to their organizations..........................25
Figure 14 - A group of users sharing the same organization are selected (orange box, top view) and all the selected users’ profiles are visualized in the User Profiles view. For each user, all the distributions of characteristics estimated from their sessions that are historically executed are listed. The orange dots are individual sessions that occurred within the selected investigation time window.................................................................25
Figure 15 - A session is selected and visualized within the Session Timeline view. ..............................................................................................................25
Figure 16 - The Diversity and Forecasting Analytics Dashboard. Top views visualize the events and the bottom views visualize the forecasting results
coming from the Diversity and Forecasting Analytics Engine as run on the selected subset.

Figure 17 - Further drilling down has been done by selecting range cells in the IP Addresses view. The rest of the views update to reflect the selection.
List of Tables

Table 1 - SECO session example................................................................. 29
Table 2 - ASM session example............................................................... 30
Table 3 - AERE session example............................................................. 31
1 Introduction

This report contains a description of the visual analytics tools developed in WP5 and their integration with various SIEM systems. The first system, the user behaviour analysis platform, consists of multiple integrated and interacting sub-components, each of them is provided as a distinct demonstrator. The goal of the platform is to analyse, model, and predict the behaviour of users. The second system, the diversity and forecasting analysis dashboard, investigates the alerts of a SIEM system in order to maximize the detection rate of fraudulent behaviour, while at the same time minimizing false alarms and to help support the evaluation of statistical models of alerts as built in other work packages. The tools are available both as code collections through a webpage and as standalone web-based applications with links to each specific component provided in the descriptions. The following chapters describe the systems and their integration with the SIEMs.

The remainder of this document is organized as follows. Chapter 2 briefly describes the visualization systems, their motivation and design principles, including improvements made according to the feedback from SIEM user partners – for more detailed information the reader is referred to D5.2 [D52]. Chapter 3 outlines the integration of the systems with a SIEM system. That is, the specific integration with our partners as well as a general description of how to integrate the system with a new SIEM. This includes a description of the requirements an application scenario has to fulfil in order for the visualization systems to be applicable. Chapter 4 concludes the deliverable with a high-level description of application scenarios in which the systems have been successfully applied in production.
2 Description of Visualization Systems

2.1 User Behaviour Analysis Platform (UBAP)

The user behaviour analysis platform is a combination of integrated tools that analyse individual aspects of user behaviour and combine the results to provide insights to SIEM operators, as well as automatically generated scores that indicate potential attacks (an overall illustration in Figure 1). We have four connected main components, including Topic modelling-based user behaviour analysis, cluster-based user modelling component and U4/VASABI visualization components. The working pipeline for the experts including three step, i.e. malicious behaviour modelling, comparative & investigative analysis of user behaviour and iterative model improvement.

![Diagram of U4/VASABI visualization components](image)

Figure 1 - Illustration of the interaction of all four components (with U4 and VASABI as separate sub-components) of the user analysis platform with a common data storage. High resolution sub-figures of the components can be referred later, including Topic modelling-based user behaviour (Figure 2) and U4/VASABI (Figure 5 and 6).

The main intuition behind user behaviour analysis is that individual sessions tracked during user’s interaction within the application partner’s system can provide insights about possible attacks or malicious actions performed. Nevertheless, direct screening of all the sessions is typically impractical due to the high amount of activity happening within a given point in time. Trying to model all the possible diverse behaviour of users can easily be highly imprecise and not capable of identifying behaviour different from normal. These points motivate uniting together for analysing not only sessions of one and the same user, but all the users that have similar behaviour throughout all their interactions with the system. This allows to build a better-informed understanding of standard user activities and be more precise when identifying suspicious behaviour.
The platform considers three major tasks that are contributing to understanding users’ behaviour:

1) The analysis of similarities in users’ behaviour in order to learn patterns of activity;
2) The modelling of regular user behaviour based on the groups of common behavioural patterns;
3) The analysis of individual and group user behaviour based both on recorded activities as sessions and on computationally extracted patterns and models

Each task is considered as a main problem by a specific sub-component of the platform; thus, the three components – Topic modelling based analysis, User modelling, U4/VASABI – together aim to analyse users’ behaviour through tight integration to provide a comprehensive solution.

The first sub-component – the topic modelling-based user behaviour analysis tool (UBAP-1) – is motivated by the first task above and responsible for visualizing the entirety of users’ behaviours and allowing to explore similarities between all the individual user’s sessions and to identify different behavioural patterns (high-level overview is given in Figure 2) as also discussed in our IEEE TVCG paper [Chen 2019]. The tool allows analysts to interact with automatically identified clusters based on the frequency of various actions that are performed by users (Figure 2). This interaction should be performed by the security or system specialists who are capable of adjusting the results of the clustering algorithm in order to better capture the similarly behaving groups of users.

LDA techniques were originally proposed by Blei [Blei 2012] to detect hidden topics from a collection of documents, each of which contains a set of words.
The input to LDA consists of a document-word matrix $DW$, where $DW_{ij}$ represents the frequency of word $j$ in document $i$ where $N$ is the number of topics. LDA produces a bag of topics $T$, a topic-word matrix $TW$, and a document-topic matrix $DT$. In our approach, we treat the actions as words and each behaviour containing multiple actions as a document. Thus, we can find hidden topics of behaviours (Figure 3), which can be seen as distinct behaviour types, or categories.

UBAP-1 presents an interactive interface for adjusting the behavior clusters. (Figure 2). It provides an overview to understand the distribution and similarity of the generated topics. t-SNE is applied to project all the topics onto a 2D space (Figure 2a), and a pie chart glyph represents each of them. By investigating and comparing the glyphs, experts can get interpretable information about the topics and assess the similarity. The projection display serves as an overview and an interactive panel, in which experts can further select representative topics.

The LDA results are collected into two matrices, the topic-action matrix and the session-topic matrix. The topic-action matrix is visualized by means of a matrix visualization (Figure 2b). The goal of the view is to enable understanding of the topic features in terms of the action probabilities. The x-axis shows the actions, while the y-axis indicates the topics. Each cell represents the action probability in a specific topic, which is encoded by the opacity. It is designed to support the identification of the key actions involved in the topics. The higher the opacity of a node, the larger the probability it is in a specific topic. The colors encode the action classes labeled by the experts (Figure 2d).

Venn diagram visualizes the similarity between topics following from the session-topic matrix (Figure 2c). The circles represent topics, and their sizes indicate the number of associated sessions. The color encodes the action class with the highest probability in the topic. Intersections between circles represent shared associated sessions. It allows experts to select representative topics. The representativeness of a topic can be concluded from the amount of intersection between the topic and others, as well as the numbers of the associated sessions (i.e., the session with the highest probability to a specific topic). The medoid topics are the highlighted stroke topics that we finally use as the output, for the LSTM in the next component.

With the feedback of one of the industrial partners (Amadeus) the component was improved in several points:

1. Provide automatically generated topics number selection and suggestions for the range of the amounts;
2. Users are able to continue the exploration of the actions’ combinations based on the suggestion;
3. Reset functions also ease the use of the system helping to clear all the custom display modifications at once.
The integration of detected user behaviour clusters for considering them in the modelling is done via the second sub-component. This *cluster-based user modelling component (UBAP-2)* builds a model for approximating the behaviour of an average user in one specific cluster. This is performed for each of the clusters identified by the topic modelling tool. After training on each of these clusters separately, this model provides predictions on the likelihood of particular behavioural aspects. The resulting models are capable of pointing out the indications of misuse in case of large divergence from expectations for a particular cluster.

Together with the industrial partner Amadeus, several improvements were integrated into the modelling process:

1. The list of actions is provided by the component user for a given use case or system, allowing for greater flexibility when it comes to applying UBAP-2 to a different use case;
2. The training process of the models (LSTM neural networks) was improved to incorporate more knowledge into them, i.e., learning to predict each action during the session, not only the last one;
3. The reported scores were changed from averaged and multiplied per-action likelihood to a perplexity score that is supposed to be more easily tractable than the previous ones.

The final task (investigating users and their sessions closely through the involvement of an expert analyst) is addressed by the eventual user-behaviour focussed visual analytics sub-components named U4 (*UBAP-3*) and VASABI (*UBAP-4*). The visual analytics tools *U4* (Figure 5) and *VASABI* (Figure 6) allow to perform an in-depth investigation of a single user’s behaviour and to compare it...
to the behaviour of other users and also to investigate the behaviour of groups of users.

The output of the models from the other components are also integrated into these tools and communicated to operators along with any other relevant data to support them in analysing potential malicious behaviour (the workflow is shown in Figure 1). To be more precise, U4 and VASABI reads through the anomaly scores produced by any user behaviour model (which can be both SKEPTIC or UBAP-2) and visualizes that data together with session information to inform the operations on the identification and analysis of suspicious users and sessions. For instance, in Figure 5, the colouring of the individual session visualizations reflect the scores coming from these models in the top menu (as also described in D5.2 [D52] and also detailed further in our IEEE TVCG paper [Nguyen 2019]).

Figure 5 - The interactive data visualization interface for U4 (UBAP-3) where the emphasis of analysis is largely on individual users and their sessions.

VASABI takes this one step further and uses the topic modelling results from UBAP-1 and makes them an integral part of the visual analysis solution as detailed in our recent IEEE TVCG paper [Nguyen 2019]. VASABI (Figure 6) is based on a framework supporting the analysis of collections of users and the numerous sessions of activities they conduct within digital applications. The framework is centred around the concept of *hierarchical user profiles* that are built based on features derived from sessions, as well as on user tasks extracted using the topics models extracted by UBAP-1 to summarise and stratify user behaviour. The approach can be summarised as:

- Utilise the topic models and treat each topic as a *user task*
- Associate sessions to different tasks
- Enrich the user profiles with these derived, latent user tasks by generating a distribution to represent the most common tasks they perform (leftmost feature on the top-right User Profile view in the interface in Figure 6)
- Cluster the users based on their task profiles (i.e., using their task distributions as a distance space)
• Visualize the clusters along with their most frequent users and most characteristic tasks (top-left corner)
• Jointly analyse users, the clusters they belong to, along with individual sessions to support a multi-level, comparative user behaviour analysis process

![Figure 6 - The interactive data visualization interface for VASABI (UBAP-4) where the emphasis of analysis is on the collective analysis of users through clusters of users (computed through a technique building on the results of UBAP-1) and understanding individual user behaviour.](image)

We envision these data-driven interfaces as the operator-facing solutions where we surface the modelling work performed in the other components and where we provide interactive and accessible representations of individual sessions and users.

Each of the components of the platform can be used separately and provide useful insights about users’ activity during their interaction with monitored applications. Therefore, the visual analytics tool displays various aspects of the current sessions, the topic modelling component helps to identify groups of users by the commonalities in their sessions, the user-modelling component can also build models for describing user’s behaviour in general. But the full potential of the platform is achieved through the interaction between them.

To achieve a productive exchange between the four sub-components, individual tools are integrated to work together on a common data storage (see Figure 1). This data storage is updated in real time and filled with logs of users’ sessions. The common storage serves as an exchange platform for the components (details are provided in the following sections) to read and write information into. At its core, our integration approach is designed to support an **iterative flow of information** within all the components. For instance, any knowledge obtained through the
analysis of individual user behaviour in U4 & VASABI can be utilised to guide the analysis of the similarities within different users’ sessions in the topic-modelling component to identify user clusters. After these clusters are stored in the data storage, this information is available for all the components and can be used to split users into common clusters characterized by similar behaviours. To exemplify, the modelling component can utilise the clusters to generate more precise and powerful models compared to models that do not take user clusters into account. Eventually, the output of these models is produced in real time together with the recorded activity and stored in the same data storage which in turn are considered by the visual analytics tools U4 and VASABI – the visual representation includes these modelling prediction results to highlight individual sessions that should be investigated more thoroughly. It is, in essence, this idea of iterative analysis and modelling that fuels our user behaviour analytics approach. More detailed information about the components can be found in deliverable D5.2 [D5.2].

2.2 Diversity and Forecasting Analytics Dashboard

This dashboard and the Diversity and Forecasting Analytics Engine component described in D6.3 [D6.3] (which is in-submission parallel to this deliverable report) form a single component that aims to provide insight into the diverse configurations of the monitoring tools and their predictive capabilities (Figure 7). The ultimate goal is to build solutions for SOC operators and cyber security analysts in helping them make better informed decisions during both the design of a defence strategy and during the evaluation of ongoing attacks and anomalies.
Figure 7 - Diversity and Forecasting Analytics Dashboard showing the events on a temporal axis along with summary views to highlight the IP and domain range, as well as visualizations to indicate multiple modelling results.

In terms of pipeline, the dashboard component receives data from SIEMs, invokes the engine to apply the predefined models, visualises the model output, and offers interactive capability for the analysts to filter data and refine the models. More specifically, three main goals that we want to support the analysis are described below.

**Goal 1 – Overview of alerts**
The alert data comes from SIEMs 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 to gain 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.

**Goal 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.

**Goal 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.
3 Integration of Visualization Systems

3.1 UBAP

UBAP consists of several technical parts: python scripts for one-time and constant background running, web interfaces for interaction with experts.

3.1.1 General Requirements

The user behaviour analysis component is required to be run on a server machine, with possibility to run python scripts, and act as a web server with a live connection to an ElasticStack where data coming from logging systems is stored and results from the UBAP subcomponents are written to.

Different sub-components have specific requirements; however, the following python packages are required to be installed within a Python environment (ver 3.6 or higher):

1) lda package
2) ElasticSearch connection package
3) Keras and tensorflow packages
4) sklearn package
5) genism
6) pandas
7) numpy

All the code for the UBAP components are openly available as an open source repository that can be accessed on this link: https://github.com/disiemprojecteu

Links to Live Demonstrators Running on Production

In addition to these, all the web-based UBAP components are accessible as live applications that are currently running on production data. We revisit these links within the discussions in this chapter, however, we list these links here collectively:

- **UBAP-1** - Topic modelling-based user behaviour analysis tool
  - **username**: ubap, **password**: BXn_*sU6iYh7MQxQ

- **UBAP-2** – U4: Session-level investigation interface
  - **username**: disiem, **password**: BXn_*sU6iYh7MQxQ

- **UBAP-3** – VASABI: User-level investigation interface
  - **username**: disiem, **password**: BXn_*sU6iYh7MQxQ
### 3.1.2 How to Install and Run

Installation requires only copying the scripts and components of web interfaces on the server. There are un-interactive functional parts:

1. LDA data download and upload script (lda_generator.py)
2. Visual analytics module (topic modelling-based behaviour analysis)
3. Behaviour models training script (training_component.py)
4. Annotator script (annotator_component.py)
5. Operational tasks extraction and user clustering (modelling.py)

In addition to these, the web-based interfaces for the various sub-components need to be on locations where the web server is able to serve them publicly. The three web-based solutions are:

1. LDA Topic modelling visualisation front-end (**UBAP-1**) ([ubap-1/lda_va/index.html](http://ubap-1/lda_va/index.html))

The scripts include configuration files in which the connection to the central database (ElasticStack) should be specified and also the index and range of data to be used for training and for annotating. Other specific details (where the models should be stored locally on the server, what is the number of clusters created after visual analysis) should be specified by technical specialists. Also, a full list of possible actions of users in the recorded logs should be included in the resources folder of the component.

**UBAP-1: LDA-Visualization**

**Setting up LDA-Visualization**

1. Connect the elasticsearch server to get the data (dump as asm-data.json in the example) and produce LDA topics based on the data:
   ```
   python lda_generator.py generate
   ```
2. Put the output files of step 2 (lda_doc_topic_final.csv and lda_topic_word_final.csv) and step 1 (asm-data.json) to lda_va/data, and run a python server in that folder:
   ```
   python -m http.server
   ```
3. Interactively explore the LDA results, and generate the category.json as output category, and run to submit the category information:
   ```
   python lda_generator.py upload
   ```

**Workflow for using LDA-Visualization**

**Note:** A live version of this interface can be accessed on:
**username:** ubap, **password:** Bxn_\*sU6iYh7MQxQ

Figure 8 - Workflow for demonstrating how to use the visual analytics tool for topic modelling-based user behaviour analysis (UBAP-1).

- The experts first examine the overview of the projection of topics (Figure 8a). They can gain idea of the overall distribution and similarity of behaviour categories.
- By brushing a group of topics in the projection, experts can examine the detailed features of the selected topics (Figure 8b). The matrix provides the distribution of actions with various probabilities and highlight a medoid as the representative.
- The expert can further separate the selected groups, adding new topics or remove the topics to update the selected results based on iterative exploration.
- The selected topics are shown in the chord visualization, showing the intersection and representativeness of the selected topics.
- After iteration, the experts can select a group of representative topics, which cover the different patterns of user behaviours and make it as input of the UBAP-2.
UBAP-2: Modelling

Setting up the Modelling Toolkit

The modelling solution do not require further setup once the steps described above in Section 3.1.2 are taken.

Workflow for using the Modelling Toolkit

The first step is to run the training component for creating models (LSTM neural networks). The script `training_component.py` is run for this once and it saves all the models locally on the server and path to get to them in the database for further use of annotation component. When the training procedure runs through successfully the annotation script can be run in the background mode (`annotator_component.py`). It will regularly check new sessions loaded to the database and use saved by the training component models to produce scores, assign them to every action and each session overall and load these scores back to the database.

UBAP-3: U4

Setting up U4

In order to integrate the U4 interface within the working environment of Amadeus, we transferred the interface to a virtual machine that is working on the clone of the Amadeus SIEM system. Given that the U4 interface is fully web-based and all the functionality is encapsulated in the front-end computation, i.e., functionality is implemented in Javascript without the need for external
computational computing sources, we were able to transport the functionality to the test environment with only modifications to data access routines (Figure 10).

Figure 10 - The data flow setup to integrate U4 and VASABI interfaces on top of live elasticsearch stack as is the case for Amadeus's setup.

**Workflow for using U4**

**Note:** A live version of this interface can be accessed on:
- **username:** disiem, **password:** BXn_*$U6iYh7MQxQ

Figure 11 - The U4 interface running on live data drawn from the ElasticSearch index for Amadeus. In this screenshot, multiple users have been selected on the top Session Overview menu and the sessions from the selected users are visualized within the Session Timeline.

The routine for using U4 is as follows:
- Analyst identifies the elasticsearch index that they are working on and sets the "Index" text on the interface accordingly.
• Since the index contains normally a wider time period and thus significant
  volumes of data, the analyst determines a time period selected for analysis
  and sets the text for “Start Time” and “End Time” (the interface element for
  this is hidden for U4 but this can be seen in the VASABI interface)
• (Optional) If there are users or organizations that are of particular interest,
  because some observation is made about them in other tools in the SIEM
  environment, those users’ and organisations’ IDs are also entered.
• Once the resulting data is fetched, the interface displays the data ready for
  further analysis (in ways described earlier in D5.2 [D52]) as can also be
  seen in Figure 11.
• Different users can be chosen on the top interface named “Session
  Overview”. More than one user can be selected by pressing the SHIFT key
  and drawing a rectangle selection on this menu.
• Once a user(s) is selected, all the sessions of that user will be listed in the
  bottom section for detailed investigation
• One can hover over the actions and the sessions to see further details and
  tooltip information

UBAP-4: VASABI

VASABI performs computational modelling at the backend including:
- extraction of common operational tasks from user actions (using topic
  modelling),
- grouping users based on the tasks they perform (using clustering analysis).

In terms of maturity, VASABI is an iteration over U4 and contains most (but
selective) functionality from U4 and adds on top the ability to do user group level
analysis.

Setting up VASABI

VASABI requires the clustering process based on topic modelling to be run before
it is ready for analysis. It requires the results of the clustering to be on the
ElasticStack or the webservice. We provide scripts that care of these operations.

When to run the modelling? The modelling step do not need to be run every time
but regularly to adapt to the changes in the user behaviour. It should also be run
when a new index is used or when new data is added to an existing index. Note
that there is no update mechanism for the modelling, which means completely
new operational tasks and user clusters will be modelled every time the modelling
script is executed.

When the script is run, it will compute tasks and clusters, then
- write the values to a new index called `vasabi` with `.id` as the name of the data
  index
- add a new field `topicId` to the sessions in the data index

Stage-1: How to run the modelling script
The Python script can be found at `src\analysis\modelling.py`
1. Install the following packages: `elasticsearch`, `gensim`, `pandas`, `numpy`, `sklearn`.
2. Edit the configuration for your index in `src\analysis\config.json`:
   - `connection`: the connection string to the data index including authentication details.
   - `index`: name of the index.
   - `size`: by default, Elasticsearch limits 10,000 results can be returned from the query. To work around this issue, the script will change that limit to the value in this key and change it back after the script is completed. So, let this key to the value bigger than the expected size of your query.
   - `filter_path` and `filter_fields`: do not change unless the current data structure has changed.
   - `start_time` and `end_time`: the duration of data you want to query.
   - `num_tasks`: the number of operational tasks you expect, say 10.
   - `num_clusters`: the number of user groups you expect, say 10.
3. Run the script such as `python modelling.py`. It could be slow if the data set is large (hundreds of thousands of sessions).

Stage-2: How to run the frontend
1. Assume that you will be setting up the VASABI tool in a folder named `vasabi`.
2. Copy 3 folders `amadeus`, `core`, `vasabi4` from `WP5\City\versions` to your `vasabi` folder.
3. In your `vasabi` folder, start a web server, such as `python -m http.server`. The VASABI tool will be available at `http://localhost:PORT/vasabi4` (replace PORT with your server port).

Once the above steps are conducted, the system should be ready for analysis – a screenshot of the interface with all the views populated can be seen in Figure 12.
**Workflow for using VASABI**

**Note:** A live version of this interface can be accessed on:
- **username:** disiem, **password:** BXn_*sU6iYh7MQxQ

Once the system is ready for analysis, the following steps are taken for conducting an investigation:

- Change the configuration accordingly with the settings of the elasticStack at the top of the visualisation then click *Fetch Data* button to retrieve the filtered data.
- Users can choose different time windows and different UserIDs (separated by commas) if any particular user ID is available. The routine workflow here could involve a cyber analyst identifying this time period and/or user IDs informed by their usual analytical tools (such as the regular Kibana based tools)
- Once the data is fetched, the first step is to select a time window for investigation and the User Overview view will be populated with the users in that investigation time window as can be seen in Figure 13.

![Figure 13](image1.jpg)

**Figure 13** - First step is to select a time window for further investigation and the User Overview is populated with user data where each bar represents a single user coloured with the average score of their sessions. In this view, the users are grouped according to their organizations.

- The analysis continues with selections of users. In the default configuration, the users are grouped according to their organization. It is possible to select either individual users or groups of users. In Figure 14, a group of users are selected from the same organization and the **User Profiles** view visualizes the list of users visually.

![Figure 14](image2.jpg)

**Figure 14** - A group of users sharing the same organization are selected (orange box, top view) and all the selected users’ profiles are visualized in the User Profiles view. For each user, all the distributions of characteristics estimated from their sessions that are historically executed are listed. The orange dots are individual sessions that occurred within the selected investigation time window.
Once a session is selected within the User Overview view (can be selected either clicking one of the orange dots or the category), the session(s) are listed in the Session View as seen in Figure 15.

![Figure 15 - A session is selected and visualized within the Session Timeline view.](image)

### 3.1.3 Integration with Partners

We integrate our visualization and analysis component into Amadeus SIEM system. The whole system works in the Amadeus network, connecting to its database with logs of the user sessions (ElasticStack). Analysts can explore the visual interface, and generate the user behaviour clusters based on LDA, retrain LSTM models, and constantly run annotator component (restarting when new models are generated) to score the sessions and warn about outlying behaviours. VASABI integrates the computed scores for its visualisations.

### 3.2 Diversity and Forecasting Analytics Dashboard

#### 3.2.1 Requirements

The Diversity and Forecasting Analytics Engine as described in D6.3 [D53] needs to be up and running.

#### 3.2.2 How to Setup

1. **Locating the Source code**
   Locate the source code package as seen in the git repository under `02-Work-Packages/WP5/City/diversity` (and also submitted as a package for review).

2. **Preparing the Data**
   - All data files should be placed in `examples/data` folder. Assume that they are in `.gz` format with the same structure as they are currently. These files need to be filtered and converted to a format that the visualisation expects.
   - Go to `src` folder, run `python convert-amadeus.py` to perform the conversion to a single output file, `examples/data/events.json`. The current script extract only events that are false positives, raised by a commercial web application protection solution used by Amadeus.
3. Running the interface
- Configure the server running RGM model. Open `config.json` in `events` folder and edit the value of `rgmServerUrl` field.
- Start a web server to host the code and data file. Go to `examples`, say using the built in Python webserver in Mac, run `python3 -m http.server`.
- Then, the visualisation should be available at `http://localhost:8000/events` as seen in Figure 16.

![Figure 16 - The Diversity and Forecasting Analytics Dashboard. Top views visualize the events and the bottom views visualize the forecasting results coming from the Diversity and Forecasting Analytics Engine as run on the selected subset.](image)

**Workflow for using the Diversity and Forecasting Analytics Dashboard**

Once the dashboard is setup, the following steps can be followed:
- A time duration is selected for focusing on the analysis and the IP addresses view and the URI distribution view updates automatically
- If a dense region in the IP range is spotted, such as the dark area on top-right in the **IP Addresses view**, further drilling down into a range can be done by clicking on individual cells, which will update the **IP Addresses view** and the **URI view** to reflect the selected events. An example of this can be seen in Figure 17 where more diversity of IP addresses used can be seen in a sub-range.
Figure 17 - Further drilling down has been done by selecting range cells in the IP Addresses view. The rest of the views update to reflect the selection.

- Once an interesting subset of events is selected, the analyst can click “Model Selected Events” which triggers a call to the Diversity and Forecasting Analytics Engine with the selected events.
- The resulting RGM based forecasting results for this particular selection of events is then displayed through the Prediction and Uplot views as seen in the bottom part of Figure 16.

3.2.3 Integration with Partners

We integrate our dashboard component together with the engine component (Deliverable 6.3 [D63]) into Amadeus testing environment. The details of the integration are discussed under Section 2.1.2 of (Deliverable 6.3 [D63]).
4 Usage of Systems in Production

Here, we provide only the specifications for the use-cases considered for the evaluation of the components within production through the integrated components. Note that we are only introducing the specification of the use-cases and the details of how these use-cases were conducted and how the validation took place is discussed in great detail within D7.3 [D73].

4.1 Use-Cases for the User Behaviour Analysis Platform (UBAP)

Sell Connect (SECO)

For travel professionals, both novice and expert who sell, advise and service leisure and business travellers, Amadeus Selling Platform Connect (SECO) is a one-stop shop for global Amadeus, and regional or local content, that is designed for the needs of the user and the business.

Selling Platform Connect is a web-based e-commerce application targeting travel agencies in particular and travel and leisure service provider in general. More details on the SECO product can be found in the Amadeus website [SECO].

Selling Platform Connect is used as a validation use case for many DiSIEM extensions, aiming to detect fraud and functional misuse and reduce investigation and intervention time.

The extensions to be validated for this use case are: Skeptic II, Vasabi and UBAP. For all extensions, the input is a set of SECO user sessions. Each session corresponds to a sequence of actions performed by a SECO user. In Table 1, we provide an example of a typical SECO user session.

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Action</th>
<th>User</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jun 30, 2019 21:58:43.432Z</td>
<td>Login</td>
<td>SECO_USER</td>
<td>User Logging in</td>
</tr>
<tr>
<td>Jun 30, 2019 21:59:43.432Z</td>
<td>DisplayFlightAvailability</td>
<td>SECO_USER</td>
<td>User Displaying seats availability for a particular flight number and a date</td>
</tr>
<tr>
<td>Jun 30, 2019 22:02:43.432Z</td>
<td>BookFlightSegment</td>
<td>SECO_USER</td>
<td>User Booking a seat in a flight</td>
</tr>
<tr>
<td>Jun 30, 2019 22:08:43.432Z</td>
<td>AddTravelerDetails</td>
<td>SECO_USER</td>
<td>User adding traveller details</td>
</tr>
</tbody>
</table>

Table 1 - SECO session example.

Amadeus Security Management (ASM)

Amadeus Logon and Security Server is the master database where security and access control data for many applications is defined. Security administrators
define and maintain this data in the Amadeus Security Management (ASM) tool, in order to authenticate users and control access to each application. LSS performs the user authentication and provides a library for applications to check the user rights. ASM is the admin UI for LSS.

More details on the ASM product can be found in the Amadeus website [ASM].

Similar to SECO, ASM is used as a validation use case for many DiSIEM extensions, aiming to detect the same suspicious user behaviours.

The same extensions will be validated for this use case: Skeptic II, Vasabi and UBAP. For all extensions, the input is a set of ASM user sessions. Each session corresponds to a sequence of actions performed by an ASM user. In Table 2 we provide an example of a typical ASM user session.

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Action</th>
<th>User</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jun 30, 2019 21:58:43.432Z</td>
<td>Login</td>
<td>ASM_USER</td>
<td>Admin Logging in</td>
</tr>
<tr>
<td>Jun 30, 2019 21:59:43.432Z</td>
<td>CreateUser</td>
<td>ASM_USER</td>
<td>Admin creates a new LSS user</td>
</tr>
<tr>
<td>Jun 30, 2019 22:02:43.432Z</td>
<td>AssignRole</td>
<td>ASM_USER</td>
<td>Admin assigns a new role to the created user.</td>
</tr>
<tr>
<td>Jun 30, 2019 22:08:43.432Z</td>
<td>AssignACL</td>
<td>ASM_USER</td>
<td>Admin assigns user to a particular Access Control List.</td>
</tr>
<tr>
<td>Jun 30, 2019 22:18:43.432Z</td>
<td>ConfirmChange</td>
<td>ASM_USER</td>
<td>Admin Confirming Changes</td>
</tr>
</tbody>
</table>

Table 2 - ASM session example.

4.2 Use-Cases for the Diversity and Forecasting Analytics Dashboard

E-Retail (AERE)
Amadeus e-Retail (AERE) is an online booking system for airlines, used for over 100 airlines. Unlike the first two use cases, AERE is targeting the end users directly. When making an online booking, users start with a flight search in an airline website, then the rest of the booking flow is handled by AERE. More details on the AERE product can be found in the Amadeus website [AERE].

The extensions that will be validated for this use case are: Skeptic II and Diverse Monitoring extensions. Both extensions aim at improving scraping activities detection and assessing the performance of different bot mitigation/detection tools.

For both extensions, the input is a set of AERE user sessions. Each session corresponds to a sequence of HTTP requests corresponding to the booking flow. In Table 3 we display an example of a typical AERE user session.
<table>
<thead>
<tr>
<th>Timestamp</th>
<th>HTTP Request</th>
<th>User</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21:58:43.432Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jun 30, 2019</td>
<td>GET '<a href="https://www.airline.com/displayFares">https://www.airline.com/displayFares</a>'</td>
<td>ANONYMOUS_USER</td>
<td>User displays flight fares</td>
</tr>
<tr>
<td>21:59:43.432Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22:02:43.432Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22:08:43.432Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jun 30, 2019</td>
<td>POST '<a href="https://www.airline.com/confirmBooking">https://www.airline.com/confirmBooking</a>'</td>
<td>ANONYMOUS_USER</td>
<td>User confirms the booking</td>
</tr>
<tr>
<td>22:18:43.432Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jun 30, 2019</td>
<td>GET '<a href="https://www.airline.com/displayTrip">https://www.airline.com/displayTrip</a>'</td>
<td>ANONYMOUS_USER</td>
<td>User displays the booking</td>
</tr>
<tr>
<td>22:19:43.432Z</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3 - AERE session example.
5 Summary and Conclusions

The deliverable described two fully integrated and operating visualization systems that are applied by our project partners - several components are already applied in production. Furthermore, the deliverable details how the systems are integrated with the partners, how they have been adapted according to the partner’s feedback, and how they can be used by others. Lastly it illustrates use-cases for which the systems are currently applied in practice. With this, the deliverable demonstrates how visual analytics and machine learning in general, and the developed systems in particular, facilitate the analysis of SIEM data. The UBAP system provides an effective tool for the analysis and modelling of user behaviour; the diversity and forecasting analytics dashboard allows for an interactive generation of statistical models for a more involved, better-informed user experience. All systems are easy to use, tested in practice and available as open source.²

² https://github.com/disiemprojecteu
References


[D63] DiSIEM. (2019). D6.2 - Use case demonstrators (in submission to EC)

[D73] DiSIEM. (2019). D7.3 – Validation results (in submission to EC)

