Sharing and Automation for
Privacy Preserving Attack Neutralization

(H2020 833418)

4.8 Demonstrator for tracking provenance in visual analyses, first version (M21)

Published by the SAPPAN Consortium

Dissemination Level: Public

H2020-SU-ICT-2018-2020 – Cybersecurity

Document control page

Document file: Deliverable 4.8
Document version: 1
Document owner: Robert Rapp (USTUTT)

Work package: WP4
Task: T4.5 Tracking of analytical provenance
Deliverable type: Demonstrator
Delivery month: M21
Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Document History:

Version	Author(s)	Date	Summary of changes made
0.1	Robert Rapp (USTUTT)	2020-01-20	Initial version of the document
0.2	Robert Rapp (USTUTT)	2021-01-27	Revised document in order to internal review
0.3	Robert Rapp (USTUTT)	2021-01-28	Improvements made on base of internal review
1	Robert Rapp (USTUTT)	2021-01-29	First version of the document

Internal review history:

Reviewed by	Date	Summary of comments
Sebastian Schäfer	2020-01-28	Grammar and spelling corrections Structure is ok Revise summary section

Executive Summary

This first iteration of the description of analytical provenance reports on the progress in the scope of T4.5 Tracking of analytical provenance in the SAPPAN Dashboard. In SAPPAN project, analytical provenance is used within the SAPPAN dashboard used by SOC agents to track the incident handling analyses performed in a SOC. After a short review of the task and its relation to other tasks, the deliverable illustrates the purpose of provenance, followed by flow charts to describe how an analysis is recorded and retrieved at a later time. In the following, the technical basis for analytical provenance and its current state of data storage and back-end implementation will be explained. After that, it is explained how layout tracking in the front-end works and how this leads to a graph-based visualisation of analysis sessions. At the end, a preview of future work is offered, including the extension of the visual representation of a session, integration of playbooks and consideration of how tracking mechanisms can be integrated outside the dashboard.

Table of Contents

1 Introduction

This deliverable describes the work performed towards completing the initial part of task *T4.5 Tracking of analytical provenance*. The deliverable itself is part of work package 4 which is about managing and automating threat intelligence. This deliverable is mainly related to the SAPPAN dashboard in work package 6, which serves as test bed for implementing tracking analytical provenance. Basic functionality of the dashboard is therefore a prerequisite for this task.

The first section is about the context of this task in the SAPPAN project. Within the second chapter, the purpose of this deliverable is first described from the perspective of analytical systems. In addition, the context described provenance in the SAPPAN dashboard. This is followed by a explanation of flow diagrams 'Record a session' and 'Replay a session' which give insights into the user interface and the communication between front end and back end. In chapter System Architecture the database model and back end functionalities are explained. This is followed by the SAPPAN dashboard implementation details in the front-end for layout tracking and the user interaction. After that a graph-based visualisation of analysis sessions is described. In Future Work the upcoming steps are explained.

2 SAPPAN Context

As given by the Grant Agreement, this deliverable should expand the functionalities of the SAPPAN Dashboard with provenance. For this reason, data that can be recorded within the dashboard is the basis for provenance. Interactions on the dashboard that occur through logged-in SOC agents can be recorded in parallel to an analysis and summarised into a representation of the analysis. This representation cannot depict the complete analysis process, as the requirements analysis showed that SOC agents use a number of different tools to analyse an incident. The dashboard does not currently offer the possibility to track user behaviour beyond the functionalities of the dashboard. A concrete use case would be opening and using an online tool such as CyberChef in a different browser tab.

An integration of external functionalities into the dashboard is not recommended due to the variety of tools, as this would mean an enormous maintenance effort for the SAPPAN dashboard. Therefore, this deliverable focuses first on the provenance of visual analyses and considers the interaction with visualisations in the dashboard. The visualisations within the dashboard provide SOC agents with a graphical form of process, network and log data used to investigate incidents.
For this purpose, provenance will be used specifically for SOC agents to record their actions and make them retrievable at a later point in time. The challenge of the architecture is to capture the actions automatically and still create a representation that is as logical and comprehensible as possible, that an agent can follow the recorded session even after the analysis. This means that SOC agents should be able to record their analyses in order to retrieve them later, to reflect on them or to communicate with other SOC agents.

In general, the documentation of the work done is time consuming and has the purpose to persist information at that point where they have been produced. One part of the

SAPPAN project is a dashboard for security operation centres to analyse machine detected incidents from a detection service of a SOC. These analyses are the base for analytical provenance and the focus is to capture the analytical procedure. Analysis sessions in security operation centres (SOC) are frequently and executed by security agents. To expand the SOC agents' tools within the SAPPAN Dashboard, as we work on a tool to record the analysis and use the recorded data to visualize the sequence of activities during an analysis. The approach of analytical provenance in SAPPAN is based on automated tracking of the user behaviour during an analysis session and a machine-based documentation of the different steps. This is an approach that allows analysis sessions to be interpreted and understood by both humans and machines and making them comparable and suitable for a wide range of applications. In this way, a generated record of the interactions performed by the user during an analysis session should help to create a better understanding of the response and recovery process. The results of a human analysis in relation to a machine-automated threat report should be recorded and made comparable in order to improve processes where necessary.

3 Related work

In the context of cyber security, analysts use a diverse range of tools to extract meaning and insight from data e. g., in order to assess the validity of an alert. Visual analytics aims to support users by combining human and machine capabilities and although much research has been done on how to improve algorithms or make visualizations more user-friendly, real-world analysis faces additional challenges. Often, different experts have to work together to solve a problem, data undergoes several transformation pipelines and the analysis workflows involve many steps to formulate hypotheses and generate insight.
Although definitions of analytical provenance vary slightly in research, it generally refers to the history of changes in data, a system, user interactions and human understanding that occur during analysis. It ranges from the history of low-level user interactions like mouse movements or clicks to higher-level insights the user derives during his workflow and how this is connected to the system. This gives provenance a wide range of purposes to accomplish, which Ragan (Ragan, 2015) and colleagues have defined as:

Recall	Maintaining and recovering memory and awareness of the current and previous state of analysis.
Replication	Reproducing the steps or workflow of a previous analysis.
Action Recovery	Maintaining the action history that allows undo/redo operations and branching actions during analysis.
Collaborative Communication	Communicating and sharing data, information, and ideas with others who are conducting the same analysis.
Presentation	Communicating the insights or progression of the analysis with those who are not directly involved with the analysis themselves, such as the general public, upper levels of management, or analysts focusing on other areas.
Meta-Analysis	Reviewing the analytic processes themselves in order to understand and improve aspects of the analysis (such as process efficiency, training efficiency, or analytic strategies)

Table 1: Purposes to accomplish with provenance (Ragan, 2015)

4 Analytical provenance in the SAPPAN Dashboard

On the basis of Table 1 the purposes of analytical provenance in the SAPPAN Dashboard are explained as followed.

Recall: To handle incidents a lot of different tools are used and it is usual, that analysis will be paused to talk with other agents about the analysis or incident. To face that behaviour, we choose an implementation of a state-base layout tracking, that the user does not have to remind the arrangement of views at the time an agent continues his analysis or replay the records of an analysis. The analysis is therefore recorded step-by-step to keep track about the current state even after the analysis.

Replication*:* The user can follow step by step the layout changes and views in that way the current awareness of the user whom recorded the session. The state of the layout shows the selection of the visualisations or views at a specific time. But an analysis session created within the dashboard can only track and record user activities in the dashboard. To provide a complete overview about the steps in analysis, the incident response playbooks (D5.7 for details) which are shared by means of the SAPPAN sharing system can be used if no recorded session is available. Recorded analysis sessions and incident response playbooks are two different formats to represent an incident response and can be used to understand a previous analysis.

Action recovery: By use of chained analysis steps to represent one analysis session, an action history is available. With the so-called session player, we provide a possibility to go step-by-step through this action history, but also to change already recorded activities or to enrich them with further information. The adaptations of an already recorded session then shows a real work flow, in which so-called dead ends also become visible. These dead ends are paths within the session that did not lead to the desired result, but are important in order to achieve a complete history.

Collaborative Communication: The dashboard is connected to the SAPPAN sharing platform and uses the shared playbooks within the SAPPAN sharing network. A playbook provides a universal way to describe an incident response. These playbooks can be used together with recordings of conducted analyses to discuss decisions with other agents. A less direct form of collaboration is reached by commenting on an analysis session. In this case, the agent optionally expands on the recording with a text message. If the recording is subsequently made available to other SOC agents, the thoughts and ideas are captured in addition to the recorded activities. This offers the possibility of indirect collaboration within other agents in the SAPPAN network.

Meta-Analysis: The basis for meta-analysis is achieved through the dashboard's integration of playbooks and previously recorded analysis sessions, which enables SOC agents to compare their own analyses and optimise playbooks and future analyses.

4.1 Types of provenance

In addition to these tasks, Ragan and colleagues also define the types of provenance information that may be collected in a visual analytics system:

Data	The history of changes and movement of data, which include sub setting, data merging, formatting, transformations, or execution of a simulation to ingest or generate new data.
Visualization	The history of graphical views and visualization states.
Interaction	The history of user interactions and commands with the system.
Insight	The history of cognitive outcomes and information derived during the analysis process, including analytic findings and hypotheses.
Rationale	The history of reasoning and intentions behind decisions, hypotheses, and interactions.

Table 2: Types of provenance information
(Ragan, 2015)

In SAPPAN dashboard, especially provenance information about the visualisation states in the dashboard needs to be collected. Regarding the fact that we are going to collect the set of commands a user executes within the visualisations, the interactions of a user will be the main provenance information. The collected commands can be used to make this data a history of visualisation states.
With the pursued approach to put as little additional documentation workload on the user as possible, instead, a mainly automatic representation of the interactions and commands is to be recorded. Therefore, a rational or insight-oriented implementation of analytical provenance is difficult to pursue. Due to the use of predominantly machine-produced data, it is not possible to justify why a SOC agent operated his way. To enable the user a way to persist additional insights he or she can make use of the optional comment function.

4.2 Flow Diagrams

Analytical provenance in the dashboard has two main flows. The first process is shown in Figure 1 and describes in detail the activities for recording a new analysis session. The second process is shown in Figure 2 how a user interacts with the dashboard to replay a session which is already recorded.

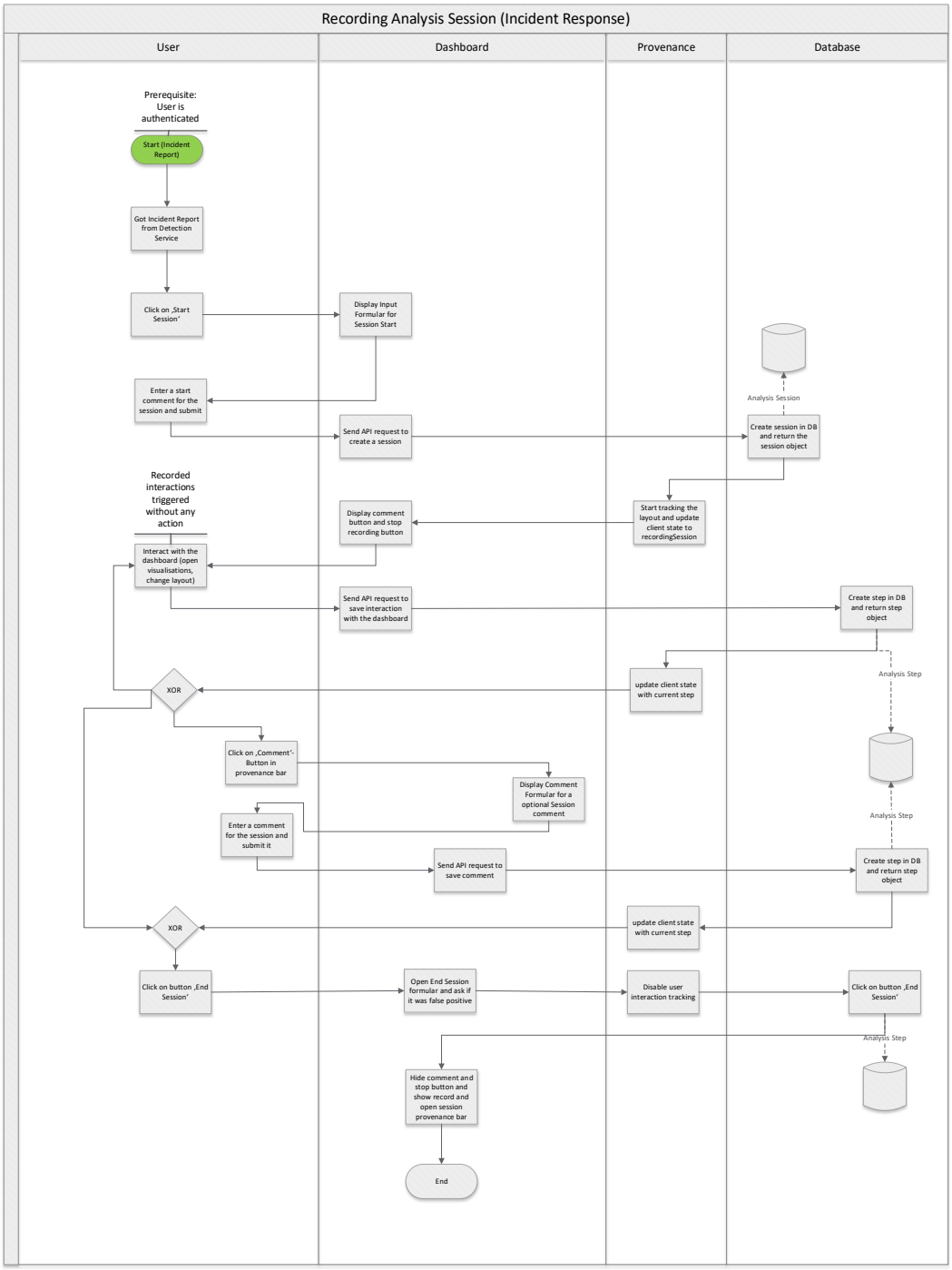

Figure 1: Flow Diagram for recording a session

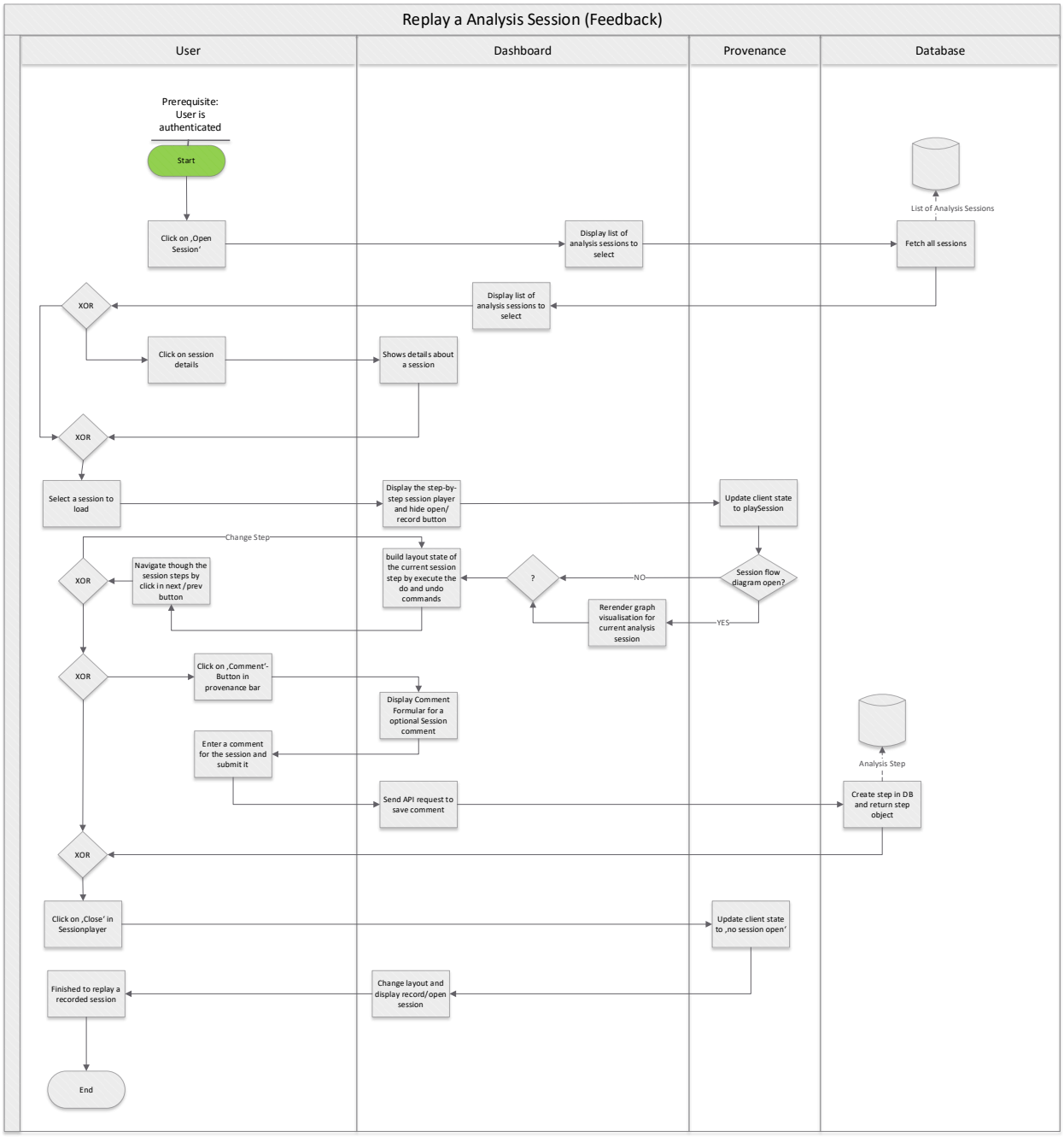

Figure 2: Flow Diagram for replay a recorded session

5 Prototype

5.1 Architecture

The system architecture for these deliverable builds on the architecture of the SAPPAN dashboard. This architecture in Figure 3 shows the dashboard architecture with its connections. Technically, the provenance tracking implementation in the SAPPAN dashboard is split between the ASP.NET Core backend and the Vue.js front end. The front end and the back end are loosely coupled through a REST API. The provenance database is used to store provenance data. This data is generated during an analysis in the front-end and is stored in the database via the back-end functions. The analytical provenance tool is shown decoupled in the Figure 3 for better understanding, but as shown below it is implemented in parts within the presentation layer, the application layer and the data storage layer.

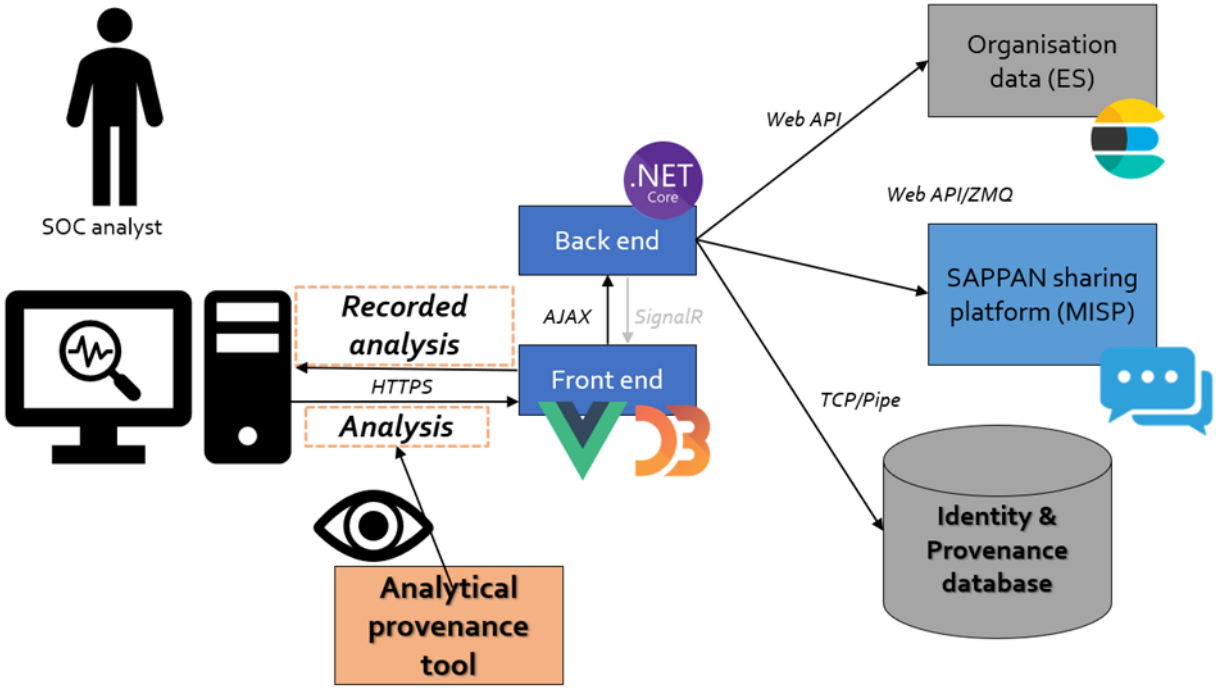

Figure 3: System architecture for analytical provenance

5.2 Implementation details

5.2.1 ASP.Net Core backend

The backend is implemented in ASP.NET Core. The backend has a REST Interface to interact with the client and a native connection to the provenance database. The backend makes it possible to pull information from our SAPPAN Sharing System. In the future it is planned to implement a push service to share information with others. In other words, MISP makes it possible to exchange information with other organisations. The sharing system give access to response and recovery playbooks from WP4. These playbooks show a way to handle an incident. The response actions which a SOC agent is executing in the dashboard can be compared with the actions in the playbook.

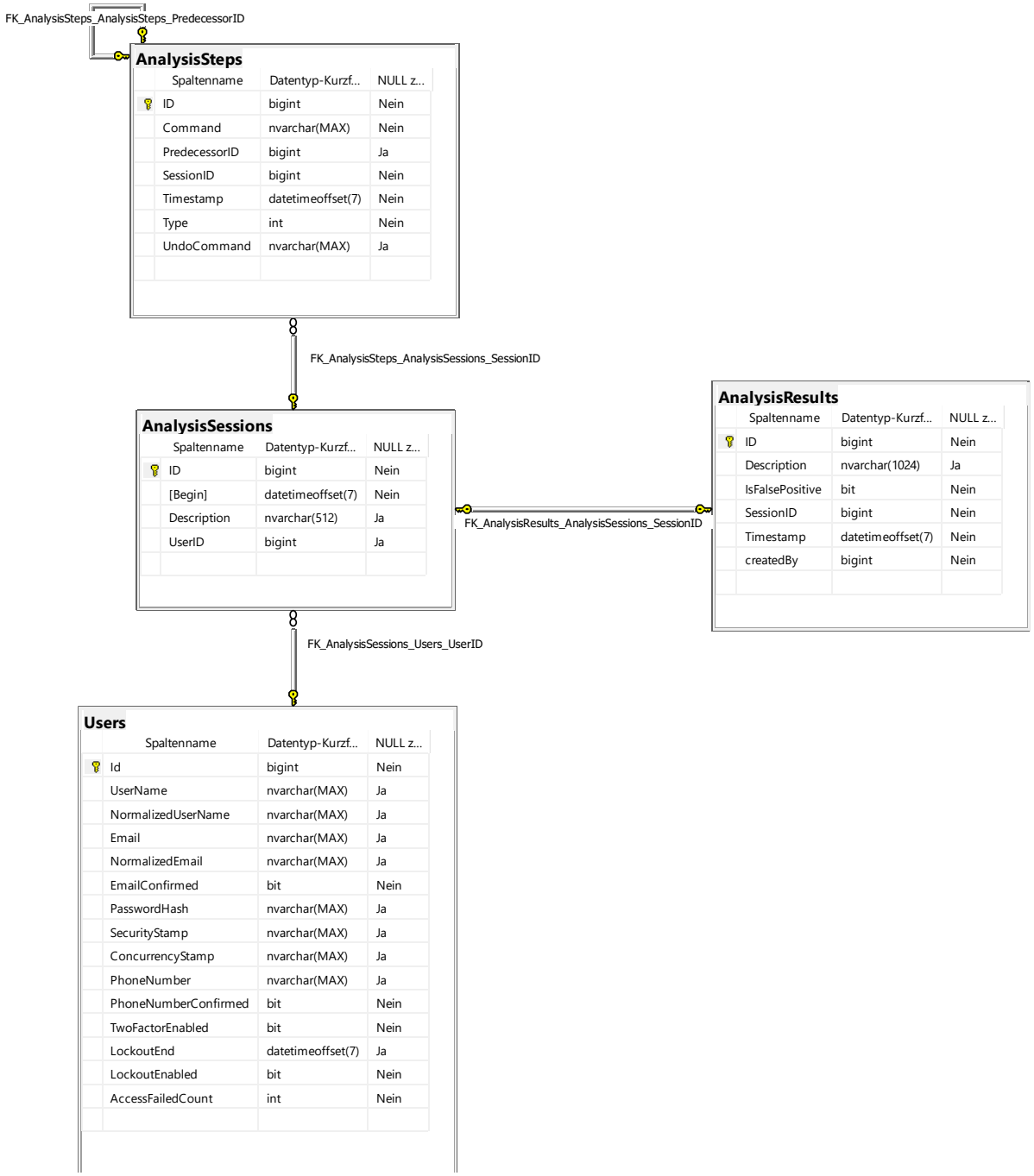

Figure 4: Entity Relationship Model of provenance database

The back end directly interacts with the SQL database which stores the provenance data we record. Depending to the data model of recorded sessions a relational database maps best to the recorded data. We based the design of our provenance tracking solution on the command pattern, that is on a collection of objects representing the actions being performed.
Figure 4 shows the entity relation model of the provenance database. The model shows four tables named Users, Analysis Sessions, Analysis Steps and Analysis Results. This is the current structure of the database.

The user table is a representation of the ASP.NET Core Identity, it adds to the SAPPAN dashboard user login functionalities. It manages common data for login and user management like user name, password, profile data and more. On the top of these columns a registration and login form can be built or used from several providers like Facebook, Google, Microsoft or Twitter. To keep track of the user activities SOC agents get a user account to identify them and make actions related to a person and for this reason a user account is necessary for provenance.
The three other tables are important for persistence of the user activities for a concrete scope, namely the recording of an incident analysis. Each session, step or result has a timestamp on its creation keep track of the sequence of actions and persist the duration of an analysis.
An analysis is always started by a user, who is then the user the analysis session belongs to. An analysis session consists of one to several analysis steps. Each step contains the action and information to which session it belongs. Due to this structure, the composition of the analysis session is flexible and can record sessions of any length, as well as continuing the recording of further steps at a later time. When an analysis is over, an analysis result is stored, which extends the session with the decision of the SOC agent whether the incident report is a false positive or a false negative. With that design it is possible to record the trigger, actions and result of an analysis. These recordings allow to replay different session states step by step and sets a visual anchor for the user to remind the current progress of an analysis even after an interruption or use of external tools. The analyst can keep track about his or her whole analysis workflow.
We based the design of our provenance tracking solution on command pattern. This is on a collection of actions-being-performed objects. In the ERM of the provenance database the implementation of specific command pattern with macro recording and an undo stack is visible. This pattern is frequently used in user interfaces where an undo stack is required, because it stores the information needed to get from one step to another. The command pattern lets toolkit objects make requests of unspecific application objects by turning the request itself into an object (Erich Gamma, 1994, p. 233). These toolkit objects are occurred by the tools in the SAPPAN dashboard like the visualisations, the layout and user dialogs and wrapped as a command in an analysis step. For example, if the user opens a specific visualisation and changes the layout, a function in the API opens the view and creates in parallel an application object which represents the change itself. This object is the basis for the creation of a new analysis step. The type of an application object gives information about what will be saved in the command and is saved in the $\boldsymbol{type}$ attribute. The types differ for example whether a user changes the layout, saves a comment or interact with a dashboard visualisation. The type is necessary for the undo stack of an analysis session, because it tells about

what kind of action the user executed and which undo command have to be executed to revert the action. Commands can be in different formats like script languages, events or text notes and the type provides information on whether, for example, a user dialogue, a JavaScript command or the arrangement of the dashboard needs to change while executing the undo stack. If a command has a backward and forward implementation, undo and redo can be implemented by storing the chain of commands. This analysis steps are chained together with previous steps by the attribute ***PredecessorID*** to realise the undo stack.

5.2.2 SAPPAN Dashboard User Interface

The SAPPAN Dashboard for response and recovery awareness is a web client for SOC agents where analysis sessions will be performed. To get a better understanding of the process shown in the flow diagrams, the SAPPAN dashboard is quickly introduced.

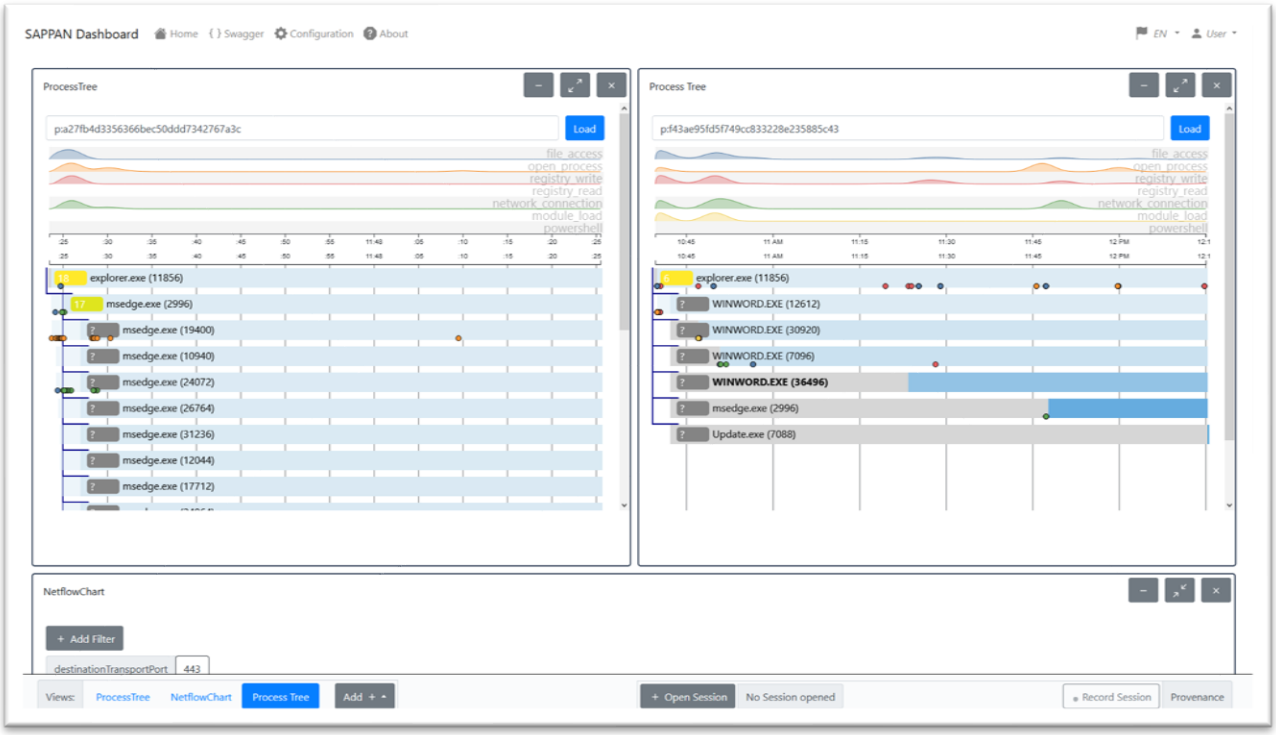

Figure 5: SAPPAN Dashboard

This user interface is a card-based grid, where a user can open, close, resize and arrange analysis views. These views visualize data from different sources like IP-NetFlow data or process data which are used by SOC agents to analyse incidents. In Figure 5: SAPPAN Dashboard at the bottom right is the Provenance bar, where the user can record sessions or view sessions that have already been recorded. To track the analyst's activity in the frontend during analysis tracking is necessary. Regarding that fact, that the recorded sessions should address professionals in IT security like SOC agents, the information captured are based on the workflow of an analysis. With

the command pattern model, the interactions of an agent in the self-implemented interactive visualisations of the SAPPAN dashboard can be expanded with tracking mechanisms and persist the interactions as commands. The commands are encoded in JSON, such that the front end can use the command parameters to undo or replay it. This makes it possible to persist the visual effects without saving resource consuming video files.

5.2.3 User interaction tracking for recording a session

- oriented to the flow diagram shown in Figure 1: Flow Diagram for recording a session

To record a session, the user has to push the record session button to open a user dialog form which requests an initial comment for an analysis session. This comment is used to describe the start of an investigation and will be saved in the provenance database.

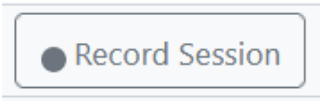

Figure 6: Record session button

If the users press on start, the user interactions with the layout are recorded. Each time the user interacts with the cards of the dashboard the changes are send to the back-end by a class called provenance service.
To track the layout, described in detail in D6.1, events are used, which emitted by card objects. The events emitted by the card objects are shown in Table 3. These events are tracked by analytical provenance functions and the current positions and size of the views is saved.

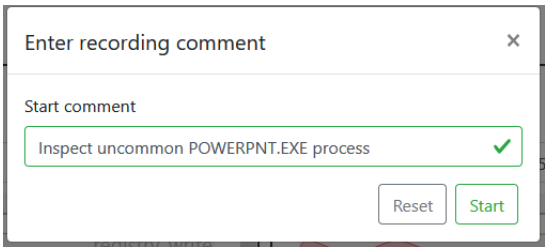

Figure 7: User dialog - Enter start comment

Name	Description
moveStart	Fires initially when an item is being moved (dragged) by human interaction
moving	Fires while an item is being moved (dragged)
moveEnd	Fires when the move is complete
resizeStart	Fires initially when an item size is changing (via human interaction)
resizing	Fires while the item is being resized
resizeEnd	Fires once resizing is complete
hoverStart	Fires when mouse begins to hover over DashItem
hoverEnd	Fires when mouse moves off DashItem

Table 3: Events emitted by a card (dash-item component)

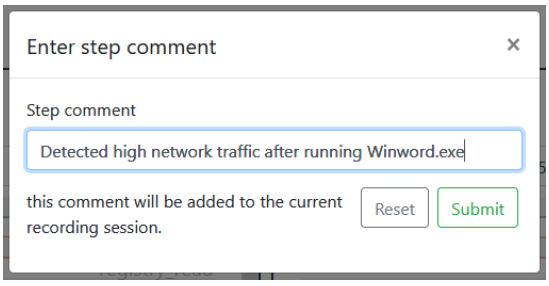

Figure 8: User Dialog

If a SOC agent wants to make a comment regarding the analysis, for example an insight, he can indicate this comment as step comment by clicking the comment button. To stop an investigation the button ‘Stop’ can be pressed. This creates an analysis result and saves it in the database. At this point the user interface stops the recording of the user activities and completes finish an analysis session by adding the related result to it.

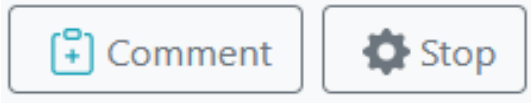

Figure 9: Comment and Stop button

5.2.4 User interaction tracking for replay a recorded session

- oriented to the flow diagram shown in Figure 2: Flow Diagram for replay a recorded session

To replay a session, the user has to push the 'Open Session'-button to open a user dialog which shows a list of sessions shown in Figure 11 to select from. This list of sessions is fetched from the backend via the provenance service and lists also sessions from other SOC agents. Beside the name and start comment of a session with a click on Details-button more information about the session are displayed. The amount of days shows the time passed since the session was created.

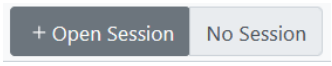

Figure 10: Open Session button and label of selected session

If the users press on open, the dashboard opens the session player in the bottom right in the provenance bar which is shown in Figure 12. This session player shows the steps of the open session and the buttons are used to navigate through the changes of the record. Each step represents a layout state and with the session player this state can be rebuild by the command and undo attribute of a session step. Beside the navigation buttons in the bottom, an interactive graph visualisation is triggered to render after the selection. This graph visualisation is called analysis session graph and should serve at this point an overview for the recorded session.

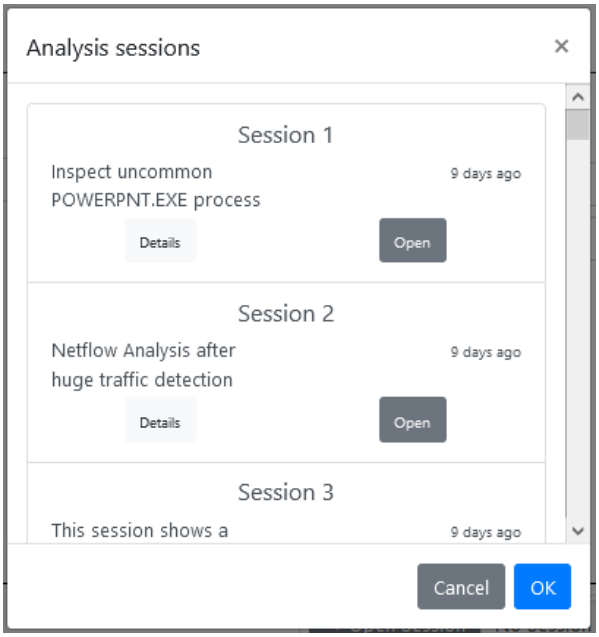

Figure 11: Analysis session list to select a recorded session

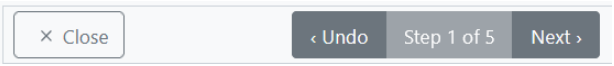

Figure 12: session player navigation area

5.2.5 Visualisation of analysis sessions

The visualisation of provenance data is depending on the data structure, how a session is recorded. As mentioned in 5.2.1 ASP.Net Core backend, a record is made with a chained command pattern which leads to the choice to use a tree graph. After comparing different graph visualisation engines, for this analysis session visualisation a graph visualisation framework in Type- and JavaScript named G6 has been chosen. G6 is MIT licensed and available on GitHub and on the G6-website. With G6 a framework was chosen, which makes interaction behaviour, animations, a set of plugins and tools available to use them for the further improvement of the visualisation. On default, actions can be triggered by clicking or hover over the graph. The visualisation has to show one to several sequences of steps as they can produce by moving back during a recording session and create a new branch in the graph. The analysis session object becomes always the root element of this graph because it shows the start of an investigation and is unique element per session. As shown in Figure 13, with the use of colours the start and end are marked as different. Each step is a child of the previous node what results in a path of steps. The end of a session is represented by the unique

element $Result$. A complete analysis session has always a path from the root node to the result node, but can have different paths which do not have to come to a result.

Figure 13: Analysis session graph first version

6 Future work

The final version of this deliverable is in M30. At this stage, the dashboard still needs to be extended to include functions that show the inclusion of interaction with layout and visualisations within an analysis session. This means that a graph can be created that represents the actual behaviour in the dashboard during an incident response. To achieve this, the visualisations need to be extended with tracking mechanisms that are used to create an analyse step. We are currently working on integrating this in the visualisation components itself such that changes to, for instance, visual appearance and filters are recorded as well. These functions must also be created for future visualisations. The session player is currently implemented for playing back an already recorded session. This will be extended so that it can also be accessed during the recording of a session. This makes it possible to replay and modify steps that have already been carried out during the session. This has the effect of helping the SOC agent to keep track of the last points at which he was working on. The use of third-party web services cannot be tracked from within the dashboard. As this information occurs in a different place, but is part of an analysis, it is essential for the traceability of an analysis session. In order to ensure the traceability of the analysis outside the dashboard, two different approaches will be considered in the future. There is the consideration of integrating the external services into the dashboard by means of deep links. Deep links would directly call up certain sub-pages of another web service and the redirection can be recorded. For the other approach, it must be checked whether the activity tracking can be integrated into external web services by means of a browser plug-in.
At this point, an evaluation of the 'Recording a session' process shown in Figure 1 can be carried out. The results of the evaluation are used to improve the user dialogues and to add missing features for the recording of a session. The session records will be used for the visual representation of the session in the session graph. The representation of a session is currently implemented as a visual component within the card layout of the dashboard. If navigation through analysis steps is also possible during recording, a live representation of the session can also be created within the analysis session graph. With a graphical representation of the session, a SOC agent can visually follow the progress of their session and click to view the steps that have already been recorded. For this use case the analysis session graph needs to be excluded from the layout tracking.

For the representation of an analysis session within the analysis session graph, the interactions with this graph must still be implemented. This should make it possible to edit and comment on an analysis session at any time. To support collaboration within

a SOC team, the graphical representation of a session provides a basis for face-to-face discussions or screensharing in video conferences. The integration of the functions comment, edit, rate and share can be considered in combination with a version history of sessions to keep track of the discussions and feedback of a recorded analysis sessions. In that case SOC agents can compare analysis sessions, give feedback or make a ranking of sessions. This is a way to enable improvements of generated analysis sessions and collect further information about a way of authoring playbooks.

As a further step, the analysis session graph will be extended so that playbooks shared via the SAPPAN sharing platform can also be used as a data source. These playbooks describe a common action sequence to handle an incident. For this purpose, the data structure of the incident handling playbook is mapped to the graphical elements of the visualisation and displayed in the same way as an analysis session. This brings about an alignment of playbooks and recorded sessions and if the functions like sharing, editing or comments are applicable for playbooks, they can be improved within the dashboard. The resulting insights can be used for the overall goal of collecting sufficient information to use the provenance data to author playbooks.

As this report details an initial version, we intend to keep working on this system.

7 Summary

This deliverable comes to its final version in M30. With the progress of D6.1 the foundations for analytical provenance have been laid. Together with the flow diagrams and use of analytical provenance in this project have come to its first implementations. The dashboard already supports recording of user-induced layout changes and the UI-functions to start, comment and stop analysis sessions have been linked to data management. Furthermore, a first visualisation of recorded analysis sessions was worked on, which shows the sessions in a graph. In the future it will be expanded with interaction possibilities and details, so that the use of playbooks from D5.7 can also be considered.

8 References

AntV, 2020. *https://g6.antv.vision/en/.* [Online].

Erich Gamma, R. H. R. J. J. V., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software.* s.l.:Addison Wesley.

Ragan, E. D. E. A. S. J. &. C. J., 2015. *Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes.* s.l.:IEEE transactions on visualization and computer graphics.

Sladden, B., 2020. *Vue responsive dash.* [Online]
Available at: https://vue-responsive-dash.netlify.app/api/#dashboard

Vue.js, 2020. *Vue.js.* [Online]
Available at: https://vuejs.org/