Data-driven insight into the puzzle-based cybersecurity training

Karolína Dočkalová Burská[a,*], Vít Rusňák[b], Radek Ošlejšek[a]

[a]Faculty of Informatics, Masaryk University, Brno, Czech republic
[b]Institute of Computer Science, Masaryk University, Brno, Czech republic

ARTICLE INFO

ABSTRACT

Puzzle-based training is a common type of hands-on activity accompanying formal and informal cybersecurity education, much like programming or other IT skills. However, there is a lack of tools to help the educators with the post-training data analysis.

Through a visualization design study, we designed the Training Analysis Tool that supports learning analysis of a single hands-on session. It allows an in-depth trainee comparison and enables the identification of flaws in puzzle assignments. We also performed a qualitative evaluation with cybersecurity experts and students. The participants apprised the positive influence of the tool on their workflows. Our insights and recommendations could aid the design of future tools supporting educators, even beyond cyber security.

1. Introduction

Higher-order thinking has become one of the essential skills for the 21st century. The best way to develop and strengthen these abilities is through practical hands-on courses [1, 2]. One commonly used learning method for training problem-solving or various IT skills (e.g., programming) is puzzle-based learning. Michalewicz et al. [3] introduced a game-based learning method that uses puzzles as a metaphor for getting students to think about how to frame and solve unstructured problems. In IT education, the puzzle-based learning approach has been prevalent for many years [4, 5, 6]. Even programming courses consist of basic concepts such as *recursion* with assignments like "Write a program to calculate the factorial of a given number."

Multiple studies confirmed the usefulness of puzzle-based learning also for cybersecurity education [7, 8, 9]. However, while hands-on training produces a tangible output in many learning areas, e.g., a code that can be checked, analyzed, and evaluated, cybersecurity training is process-oriented. Puzzles are tasks like "search for a vulnerability on server X" that are difficult to track. Tutors have only a limited view of what trainees are doing in the computer network and how they deal with the task, making the post-training evaluation challenging. This paper presents results of cooperation with cybersecurity education experts that led to the design of a visualization tool supporting the follow-up learning analysis of the training sessions.

Regardless of the education subject, tutors make intensive efforts to create, organize, and continually improve these so-called *blended courses*[1]. Trainees' assessment, which usually follows the training session, is integral to the teaching process. The focus lies on comparing individual trainees and analyzing their progress or discovering weaknesses in the training design.

We contribute to the state of the art of applying visualizations in education practice with: (a) a user requirement definition on support tools for tutors of the hands-on puzzle-based learning

---

*Corresponding author.

E-mail addresses: burska@mail.muni.cz (Karolína Dočkalová Burská), rusnak@ics.muni.cz (Vít Rusňák), oslejsek@fi.muni.cz (Radek Ošlejšek)

[1]Blended courses combine computer-supported learning activities with traditional face-to-face interaction during training sessions.

activities (in the cybersecurity education context); (b) design and implementation of the visualization tool for the post hoc analysis of data from the training session; and (c) an evaluation with domain experts resulting in design recommendations for future work.

2. Related work

Assessing the effectiveness of game-based learning poses a significant challenge in the learning analytics research domain. Loh [10] distinguishes between "assessment *for* learning" and "assessment *of* learning." The former is designed to assess a learner's understanding at the course end. The latter is more helpful to educators because it helps them to improve the learning processes. This paper deals with educators' insight into the learning process. A considerable effort has been made in the past to conceptualize data mining and digital assessment for serious games so that generic learning analytics principles can be researched and applied regardless of the specific game content [11, 12, 13]. Our solution deals with event logs and the score-based assessment that represent broadly accepted types of telemetry and evaluation data for serious games.

Our work lies at the intersection of education, visualization, and HCI research. According to the classification provided in [14], this paper addresses visual data analysis tasks of organizing participants (referred to as *tutors*). Using information technologies in blended courses enables us to collect metadata produced by learners. Tutors can use them for a post hoc analysis of learners' progression and content revision. Nevertheless, the design and deployment of efficient support tools remain a challenging problem [15]. There are general tools that could be used for specific post-training tasks, e.g., comparing score-based assessment settings via the LineUp application [16]. Our tool aims to reflect the well-defined requirements of training designers and tutors, providing them with a domain-specific comprehensible analytical dashboard.

The purpose of the post-training learning analysis is to understand and optimize learning processes. Previous works [17, 18, 19, 20, 21] address using visual dashboards for learning analysis and confirm the need for insight exceeding simple summative feedback [22]. Apart from focusing on the learning process, learning analytics in higher education also provide valuable teaching or research resources [23]. Analytical tools can support decision-making and improve pedagogical approaches.

Most of these learning analysis tools focus on the high-level perspective evaluation of students' performance. Existing surveys overview and analyze learning dashboards either for tutors [24, 25, 26] or students [27]. Most of them are related to the uptake of massive online open courses. These tools focus on visualizing learning activity, tracking specific learning goals, and providing a high-level perspective on learners' progress. Moodleboard [28] is a decision support tool for pedagogical engineers and administrators providing both course statistics and detection of flaws or misuses for an open-source learning management system Moodle. LISSA [29] aims at improving student-advisor dialogue during face-to-face consultations. The tool provides an overview of study progress or peer comparison among multiple students. SAM [30] is a general-purpose

web-based environment visualizing learners' activities, improving awareness, and supporting self-reflection. Such high-level tools represent domain-independent systems to gather, process, and report the collected and derived data while overlooking disciplinary knowledge practices.

In contrast, tools for lower-level data analysis from practical courses often require considering insight from domain experts because the input data driving the analytical tools are domain-specific. Examples can be found for math [31], where the system tackles the understanding of selected math functions, programming tools [32] that utilize compilation processes and software quality metrics for assessment, or penetration testing [33] based on knowledge graphs. Figure 1 categorizes these tools in two axes: x-axis – single or multiple training sessions; y-axis – data specificity, i.e., from the domain-specific data to derived data and metadata.
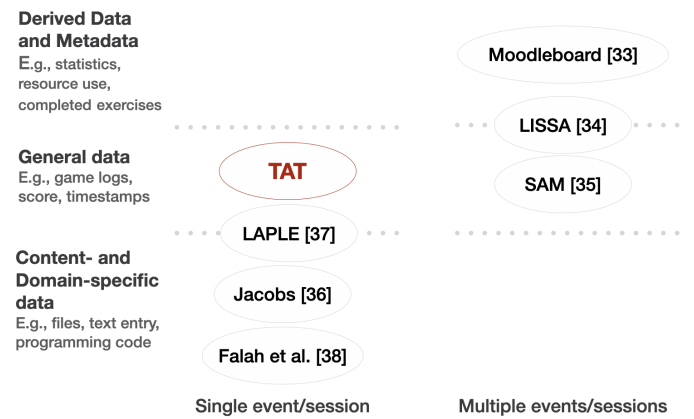

Fig. 1. Categorization of learning analytics tools based on their focus (on single or multiple sessions) and the input data types (from domain-specific to derived meta-data). TAT position is highlighted.

We propose the *Training Analysis Tool* (TAT) – a dashboard-like tool for tutors providing data-driven insight into a training session through several linked visualizations. The TAT supports tutors in low-level learning analytics tasks such as inspection and comparison of trainees or identifying training design flaws based on the data from single training sessions.

3. Background

The puzzle-based learning in the cybersecurity domain is primarily represented by *Capture the Flag* (CTF) games [34, 35, 36]. CTF training scenarios serve as puzzle-based templates structuring the content into levels focused on solving cybersecurity tasks, e.g., scan the network, identify a server, find the server vulnerability, exploit it, and gain the root privileges. CTF games can be organized in diverse ways. Very popular are unsupervised online games when a trainee can access the game or interrupt it anytime. Tutored (or supervised) training sessions for small groups are often practiced in a formal cybersecurity education or professional training. The supervised training sessions share the principles of blended courses popular in primary and secondary education.

CTF games contain a short background story, task assignments, their evaluation, hints, and solutions for each level. A typical scenario consists of up to ten levels. Finding a level solution is necessary to proceed to the next one. Training scenarios use multiple gamification characteristics such as scoring, level-based approach, or scoreboards. Trainees are penalized when taking hints or solutions and reach score points for successful solutions.

Hands-on cybersecurity training is often organized in so-called cyber ranges. The *KYPO Cyber Range Platform*[2] (hereafter referred to as *KYPO CRP*)that we use for development and evaluation is a cloud-based environment providing features for the virtualization of computer systems and networks [37]. It serves as a platform for practical training of various cybersecurity skills in university courses as well as for the training of practitioners from institutions outside. The *KYPO Cyber Range* allows us to create so-called *sandboxes* – isolated computer networks consisting of multiple virtual machines for several dozens of trainees (the exact number depends on the cloud capacity and resource requirements). The web portal provides a user interface for the management of sandboxes, users, *training scenarios*, and organizing training sessions.

A typical training session is organized for 15–20 participants in the IT classroom. Trainees log in to the web portal and launch a training scenario consisting of a sequence of cybersecurity puzzles. Trainees solve the puzzles individually in their private sandboxes without affecting others' work. A successful solution of the puzzle yields a short string (called *flag*). Entering the flag in the web portal opens the next level. Trainees who are struggling can use hints specific for each level. When helpless, they can see the correct solution (a list of steps leading to the flag). Time for solving all the levels is usually limited to the class length (one or two hours). Tutors walk around and help trainees either on request or when they realize that someone significantly lacks behind (typically by quick peek on their displays or asking them directly). In the end, the scoreboard shows individual scores, and tutors hold a short debriefing to present correct solutions.

Figure 2 illustrates the principal elements and actions of the whole workflow.

There are two broad use cases for the post-training analysis: (a) a comparison of trainees and (b) training scenario improvements. The former is essential when the CTF games are part of the competitions or exams. The rank or grade is then based on the final score and time. However, the tutor cannot understand the subtle difference in the trainee's behavior or expose cheating. Likewise, *training scenario improvements* were usually based on error-prone manual processing of the logged data and anecdotal evidence from training sessions, making revisions inefficient.

3.1. Data description

Hands-on CTF games provide two datasets available for visual analysis: a *training scenario* and timestamped *trainees'*
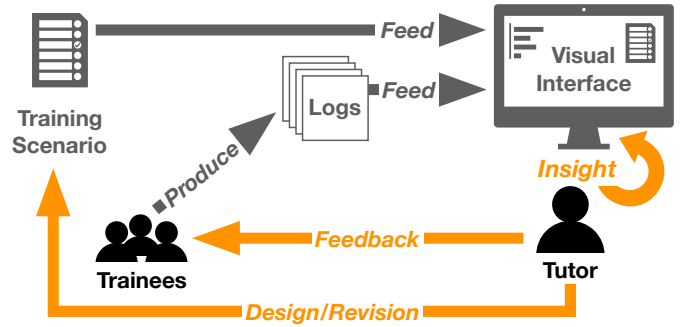

Fig. 2. The generalized training workflow. The tutor uses the visual interface to get insight into the training session (to help trainees in trouble) and to revise and improve the training scenario. The data sources are activity logs of the trainees and training scenario description which provides context.

events recorded during the training session. The *KYPO CRP* provides REST API to access these data on-demand in JSON format.

The **training scenario** contains attributes related to the content. Namely, a background story, puzzle assignments, hints, hint penalties, solutions, solution penalties, correct flags, flag score points, and level time limits. These attributes do not change during the training session. However, tutors might edit them afterward based on trainees' feedback or outcomes from training session analysis. Typical changes include fixing typos and improving the clarity of puzzle assignments, or adjusting level duration estimate, score, and penalty points.

The **trainees' events** are automatically collected when trainees interact with the web portal. Example events are: training started, training ended, level started, level ended, correct flag entered, incorrect flag entered, hint taken, solution taken. Each event contains a standard set of attributes (timestamp, event type, training description ID, training session ID, user ID). Three event types (an incorrect flag entered, a hint used, a solution displayed) contain specific attributes – an incorrect flag string and penalty points.

Although the input data is domain-specific, we can find similarities also in other forms of puzzle-based gaming. Data types are either integers (score and penalty points, level duration estimate – representing minutes) or text strings (plain-text for flags, markdown markup for all the rest).

4. Process and methods

We closely collaborate with domain experts (cybersecurity educators) from our university who represent target users. They provided initial requirements, gave us feedback on proposed designs, and participated in both evaluations. Our goal was to improve the workflow of tutors and organizers of hands-on cybersecurity training sessions through the design and deployment of the Training Analysis Tool that processes data from the *KYPO Cyber Range*.

In this project, we applied the user-centered approach guided by the design study methodology framework [38], reflecting its *core* stages: discover, design, implement, deploy. Our iterative

---

[2]`https://kypo.cz`

process has four phases. Each phase reflects one or more of these stages:

Problem characterization *(discover)*: We conducted semi-structured interviews with three domain experts from the university cybersecurity team. All of them partake in educational activities as seminar tutors or lecturers, and they also participated later on in the evaluation. Each interview lasted about an hour. We also did four field observations during training sessions to gather user requirements and complement our notes, each lasting up to two hours. From these data, we elicited functional requirements and design decisions for both tools.

Early prototype and formative evaluation *(design, implement, deploy)*: We created the early prototype and performed a qualitative formative evaluation with five collaborating cybersecurity educators and one student familiar with the CTF games.

Late prototype and summative evaluation *(design, implement, deploy)*: We added new features and redesigned the user interface based on received feedback. A qualitative summative evaluation with eight participants served us for the validation of the final designs.

Final deployment *(implement, deploy)*: The last phase includes the integration of TAT into the *KYPO CRP*. We also plan to collect further feedback from its routine usage. Unfortunately, due to the COVID-19 pandemic, the number of training sessions has been severely limited.

5. User requirements

Post-training session evaluation provides many opportunities for tutors to perform a detailed analysis of a training scenario and assessment of the trainees. The interviews and field observations revealed that tutors struggle with analyzing the training data from the individual sessions. They expressed the need for an overview of the data collected during the training session, which enables them to: analyze trainees' behavior, compare their performance, and revise the content and configuration of the training scenario.

We organized the requirements into the four main categories:

R1 – Trainee behavior analysis: Tutors should examine trainees' behavior and identify outliers – e.g., those who are extremely slow/fast or gave up the training. They should assess the trainees by comparing their results (e.g., final time and score, taken hints, number of entered incorrect flags). It is also relevant when the training session is a part of some competition. Further, reviewing the trainees' actions, such as many partially correct flags submitted by several trainees, can point out flaws in the puzzle assignment.

R2 – Assessment revision: Correctly set scores and penalties are crucial for the gameplay and trainees' motivation to complete the training. Setting the penalties for hints too small, for instance, can demotivate trainees in attempting to find the solution by themselves. Instead, they could take all hints immediately, which would even result in a better final score. Therefore, the tutors should be able to review the assessment criteria of the training session.

R3 – Timing revision: Proper estimation of time requirements for cybersecurity puzzles is tricky. Short time allocated for a challenging puzzle can delay the whole session, put unnecessary pressure on trainees to take hints early, or force tutors to intervene prematurely. During the interviews, even the most experienced tutors admitted that they do not have a proper first estimate of mapping puzzle difficulty to time limits. Therefore, tutors should be able to review the time limits of the training session.

R4 – Training content revision: Tutors should be able to analyze problematic parts of the training content to improve its quality iteratively. The trouble can be hidden either in individual puzzles (e.g., unclear puzzle assignment, useless hint) or their interconnection (e.g., the unbalanced difficulty of two successive levels).

6. Early design

The main goal of the *Training Analysis Tool (TAT)* is to display data from a single training session in the context of the corresponding training scenario (e.g., puzzle assignments, scoring, timing). The tool is designed as a dashboard combining several linked views. Its design follows principles formulated by Oslejsek et al. [14]:

- Analyze the *impact of tutor's supervision:* The tool consists of temporal views of trainees' actions and the score development at various levels of detail. Tutors can analyze the impact of both individual and class-wide interventions by focusing on the time of intervention.

- Analyze *quality of training exercise:* All views display the score and time limits that form the primary assessment criteria and delimit the training session's difficulty. These visual artifacts help tutors to analyze the quality of training. Moreover, predefined parameters (penalties, time limits, tasks) are available in the dashboard together with runtime data, enabling tutors to reveal possible weaknesses in training scenarios by comparing expected versus actual development.

- Analyze *behavior analysis of trainees:* The training session is captured from several perspectives: temporal view on trainees' activities, a static preview of final results, and detailed dynamic score development. By combining these coordinated views, tutors can interactively analyze individual trainees' behavior, compare them mutually or concerning expected behavior, and visually identify outliers.

The early prototype of the *Training Analysis Tool (TAT)* (Fig. 3) is a web application consisting of three interactive visualizations: TIME-SCORE OVERVIEW, TRAINING OVERVIEW, and INDIVIDUAL TRAINING WALKTHROUGH.

The former two are based on visualizations proposed by [39] for player-centered reflection and CTF game results. Since their input data is similar (timestamped events), we used its core design principles and visual encoding, but our visualizations provide extended interaction capabilities. We further elaborate on the design of individual TAT components in detail.

All three visualizations of the early prototype use a fixed color scheme. The colors were meant to distinguish individual levels of training and were selected in different intensities to be distinguishable for people with the most common forms of color vision deficiencies.
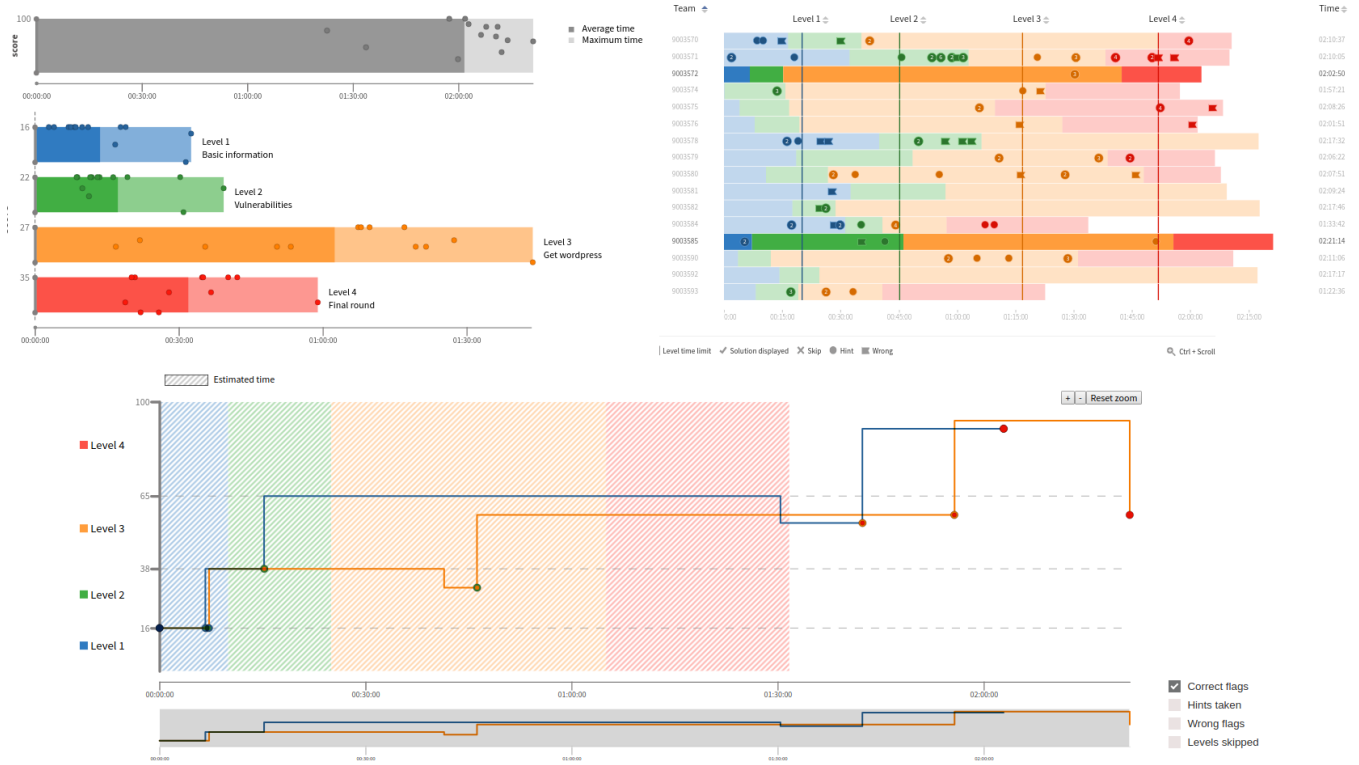
Fig. 3. The early prototype of the Training Analysis Tool (TAT) consists of three interconnected visualizations. The TIME-SCORE OVERVIEW (top-left) presents the distribution of achieved scores (final and per-level) for each trainee. The TRAINING OVERVIEW (top-right) displays the overall training duration for each trainee and their activities (e.g., taking hints, inserting incorrect flags). The INDIVIDUAL WALKTHROUGH (bottom) is suitable for a detailed comparison of two or more trainees.

6.1. Time-score overview

Total duration and the final score are two main factors used for measuring the performance of the trainees. The TIME-SCORE OVERVIEW (Fig. 3, top-left) helps identify the correlations between these factors, providing a view on the score distribution, pinpoint the outliers, or allocate clusters.

Using simple standard statistical views, such as boxplots, would be inconvenient because we need to put in the context multiple metrics (average, and estimate times, final scores). Therefore, the visualization combines bar charts with scatter plots to incorporate time and score data into a single view. The top bar shows the total time (*x*-axis) and each trainee's final score (*y*-axis). The smaller bars below represent individual levels (i.e., tasks). Each bar's length expresses the maximum time for the given level (i.e., the time of the slowest trainee). The average time is on the border of two color shades. Although the scoring span can differ in each level, the bars have fixed heights. The vertical space is sufficient to display and analyze achieved score distribution regardless of the scoring span. The maximal level or game score is on the *y*-axis, and the exact score numbers are provided on-demand as tooltips of individual dots together with a trainee's name.

Hovering the mouse cursor over the dot highlights the corresponding results of the trainee in the remaining levels highlight and the exact time and the achieved score for the level display. A mouse click on the dot highlights the corresponding data in the TRAINING OVERVIEW and displays detailed score development

in the individual training path at the bottom. Dot clusters can visually indicate the correlations between time and score, which is particularly helpful when the tutor aims to identify the training design issues such as a level difficulty compared to its duration.

The tutors can use it to analyze the results of individual trainees and put them in the context of the training group (**R1**) or to review score-based assessment (**R2**). Bar charts also help the tutors review time requirements (**R3**). Dot clusters may help in the identification of problematic levels in the training scenario (**R4**).

6.2. Training overview

The TRAINING OVERVIEW (Fig. 3, top-right) provides a detailed yet compact and uncluttered view of the trainees' progressions and activities. It is based on a stacked bar chart where each row corresponds to one trainee. Segments represent training levels and encompass related game events as glyphs. A user can filter the data based on the level duration and zoom the view to unfold the aggregated events (numbered circles) performed quickly.

The visualization shows the relative time of the training. The stacked bars are aligned to the left, so it is possible to compare the time requirements regardless of the delays caused by individual trainees' various starting times (**R3**). Level labels above the bars support sorting by the duration of the corresponding levels. The related vertical lines indicate the expected level duration. When sorted, they also reveal the deviation of the actual and estimated time for each trainee.

The glyphs indicate events. In this view, they help the tutors to recognize possible problems in the design of training definition (**R4**) or analyze the behavior of the trainees (**R1**). For example, multiple incorrect flags submitted by diverse trainees can indicate unclear or ambiguous instructions; many hints taken in quick succession may suggest a lack of effort caused by improper difficulty.

6.3. Individual walkthrough

The INDIVIDUAL WALKTHROUGH (Fig. 3, bottom) is based on a step chart with glyphs representing trainees' actions. It enables the tutors to track outliers' behavior (**R1**) and explore the cause of recognized problems in the training session (**R2**, **R4**). It provides a detailed insight into a trainee's advancement and actions or allows comparing two or three trainees selected from the TRAINING OVERVIEW list or the TIME-SCORE OVERVIEW. The y-axis represents gained score. The horizontal dashed lines imply the maximal level score. The striped background outlines the estimated level times.

A zoom function allows adjusting the view on a selected portion of the chart, which is useful when the events are clustered. On mouse hover, a tooltip shows details for each action. A context view frame below the main chart helps the tutor to get oriented in the zoomed area and shift the time range when needed. Furthermore, the checkboxes in the bottom right corner allow filtering the event types.

7. Formative evaluation

The main goal was to gain feedback on the TAT's usefulness in four areas:

- **Trainees** – Is it possible to identify trainees who struggled (e.g., lacking behind, stuck with the task/level)? Can tutors recognize any unusual behavior of trainees (e.g., cheating, prolonged inactivity)?
- **Training session** – Is it possible to recognize when the training is running out of schedule? Can tutors identify scenario design issues?
- **Visual encoding** – Is the visualization easy to understand? What type of information is redundant or missing?
- **Interaction** – How do tutors interact with the visualization? Are the interaction capabilities sufficient?

We further evaluated the usability and usefulness of the visualizations and gathered remarks on visualization improvements for the following design process iteration.

7.1. Participants

Due to the necessary background knowledge of hands-on cybersecurity training, we conducted a qualitative user study with five domain experts (P1–P5) and one student (P6). All of them were members of the university cybersecurity team who partake in hands-on training on different positions. Table 1 shows their demographic information.

Table 1. Demographic summary of the participants and their involvement in the design study. TE – teaching experience (in years), OE – organized hands-on exercises (in sessions). Participation in individual stages: PC – problem characterization; FE – formative evaluation; SE – summative evaluation.

ID	Age	Position	TE	OE	PC	FE	SE
P1	33	Lecturer, Manager	4	>20	✓	✓	✓
P2	27	Seminar tutor	7	<20	✓	✓	✓
P3	31	Seminar tutor	3	>20	✓	✓	✓
P4	27	Seminar tutor	5	<10		✓	✓
P5	35	Senior lecturer	5	>20		✓	✓
P6	24	CTF Course graduate	0	1		✓	✓
P7	22	CTF Course graduate	0	1			✓
P8	21	CTF Course graduate	0	0			✓

7.2. Procedure

In September 2019, we held the formative evaluation sessions in person using 27" iMac with the resolution 2560×1440 and Google Chrome browser version 76. The experimenter took notes and audio recorded the participants' opinions and thoughts.

The user study had two parts, and the participants were asked to think aloud. The sessions lasted about an hour. In the first part, the experimenter outlined the procedure. The participant consented and filled the demography questionnaire. The experimenter presented the TAT and situated the participant in the role of a tutor using the tool. Next, the participant spent 2–3 minutes familiarizing with it using dummy data followed by completing three tasks addressing requirements R1–R4:

- *T1: Identify an unusual behavior of trainees and name the potential issues.*
- *T2: Find and compare a pair of trainees who: a) have the same score; b) were the best and the worst; c) were the slowest and the fastest. How do they differ?*
- *T3: Identify problems caused by the poor design of the training scenario and propose improvements.*

Participants performed the tasks on two data sets DS1 and DS2. We chose the genuine data since they contain various actions observable during training sessions (e.g., guessing the correct flag, prolonged inactivity, varying trainees' performance). Their different size, number of trainees, and duration show two distinct yet ordinary real-world circumstances.

DS1 is from the tutorial on computer forensic skills and consists of six game levels. The goal is to identify and examine malicious software running in the computer system. The trainees learn how to identify a suspicious application, dissect its executable, and process memory. The session lasted 55 minutes, and 16 trainees generated 374 events, making the 23.4 events per trainee on average. DS2 is an attack-oriented training scenario that consists of four game levels with the following puzzles: exploit server vulnerability, gain the root privileges, access a protected data file, and cover the traces after the attack. Six trainees generated 146 events over 90 minutes, averaging 24.8 events per trainee.

Finally, the participant filled two usability questionnaires and was debriefed. We chose the SUS – System Usability Scale [40] and the SEQ – Single Ease Question [41], two widely used questionnaires for measuring various products' usability. The

former is a widely used method for assessing the usability of the systems. The latter is considered a robust measure to quantify the usability for tasks that are too complex for metrics like task duration time or completion rate[3] and when the number of participants is low, as in our case.

7.3. Results

The formative evaluation revealed weaknesses in the early design and helped us understand tutors' work after the training session.

The most acclaimed feature of the TRAINING OVERVIEW visualization is the ability to sort trainees by the time spent at some level and compare them to the estimated level duration (defined in the training scenario). Participants also used the visualization to identify the trainees who significantly exceeded the estimated level duration time.

For most of the participants, the SCORE OVERVIEW visualization was a starting point when solving all the tasks. They used it to identify outlying trainees (P2, P4, P6), to assess the difficulty of each level based on the time/score distribution of trainees (P1, P2, P4, P6), or to compare it with the maximum score per level (P3, P4). P3 also used the score overview to assess the conceptual design of the training scenarios (the first levels should be manageable and short compared to the final ones). Participants lacked information about estimated level duration (P1–P4, P6). P6 wanted even more details, such as medians of time and score for each level.

Participants often used score overview visualization to highlight trainees in training overview and vice versa. Score overview was also often used in T2 as a selector for trainees to compare. We did not observe any other extensive mutual use of two or all three visualizations. On the other hand, the INDIVIDUAL TRAINING WALKTHROUGH visualization was generally considered "useful only in a specific case when the training session is organized as a competition to decide the final order of trainees" (P4).

The main complaint (mentioned by all) was the absence of a tabular view showing various details of all trainees such as their final score, scores per level, number of taken hints, or incorrect flags.

Other frequent issues were: the absence of filtering features (P1–P5); a missing overview of the training scenario allowing the users to skim through the texts of tasks, set penalties, and flags (P2–P4, P6); insufficient integration of the visualizations (P1, P2, P4, P5); and the visual encoding (P1, P3, P4, P6) considered by P3 as "disturbing due to many colors without proper meaning."

The SUS score was 65.4 points (out of 100). It corresponds to the *good* rating, according to the adjective ratings [42]. Fig. 4 summarizes the SUS questionnaire responses. With the SEQ score of 6.5 (out of 7), the TAT showed to be well-suited for training design analysis (T3: Identify training design issues.). The two tasks focused on identifying and comparing trainees

---

[3]The user responds to a single precisely-worded question ("Overall, how difficult or easy did you find this task?"), using a scale from 1 (Very difficult) to 7 (Very easy).

scored 5 (T1: Unusual behavior of trainees) and 6 (T2: Comparison of trainees).
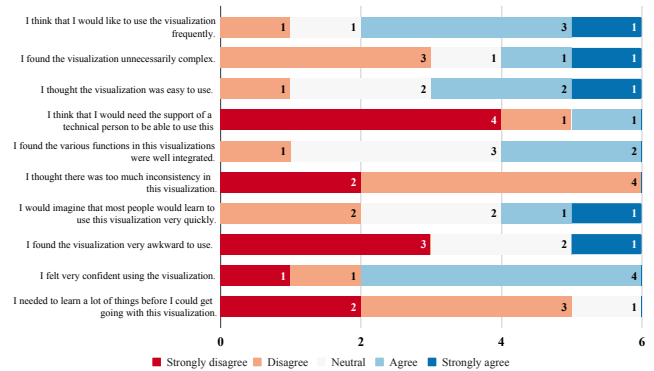

Fig. 4. Formative evaluation: The SUS questionnaire responses.

While these results confirmed the overall usability and usefulness of the TAT, we had to address the main issues raised by the study participants.

8. Final design

We revised the final design rationale, visual encoding, and interaction capabilities of the current version of the TAT based on the formative evaluation. The prototype, implemented using Angular and D3.js library, is available at `https://tat.surge.sh`.

The main principles of the three visualizations remain the same. However, we significantly redesigned the layout making the TRAINING OVERVIEW the most prominent visualization. We also added more filtering options for selecting individual trainees and revised the use of colors. The formative evaluation also revealed that the coloring of levels is not essential for the users, so we have changed it in the late prototype: the platform on which the training sessions take place generates a unique avatar for each trainee. Therefore, we decided to emphasize the trainees based on the avatar's color instead. Now, each trainee has a unique color in all three visualizations. These colors are not intended as the exclusive means of trainee identification but as complementary visual support (to accompany the ability to highlight or filter the trainees). To distinguish training levels, we used gray color shades in the late prototype.

Finally, we added additional information regarding the training definition, such as the task descriptions, correct flags, and contextualized trainees' data with individual levels. Fig. 6 displays the final layout, with the collapsed TRAINING DEFINITION SUMMARY and VISUALIZATION FILTERS sections.

8.1. Training definition summary and visualization filters

The TAT's upper part (Fig. 5 – A) contains a collapsible panel with the training definition details, visualization filters, and avatar-based trainees filter. The TRAINING DEFINITION SUMMARY serves for the configuration of the tool and synopsis of the training. It provides training scenario parameters (i.e., task assignments, hints, penalties, correct flags). The tabs show data
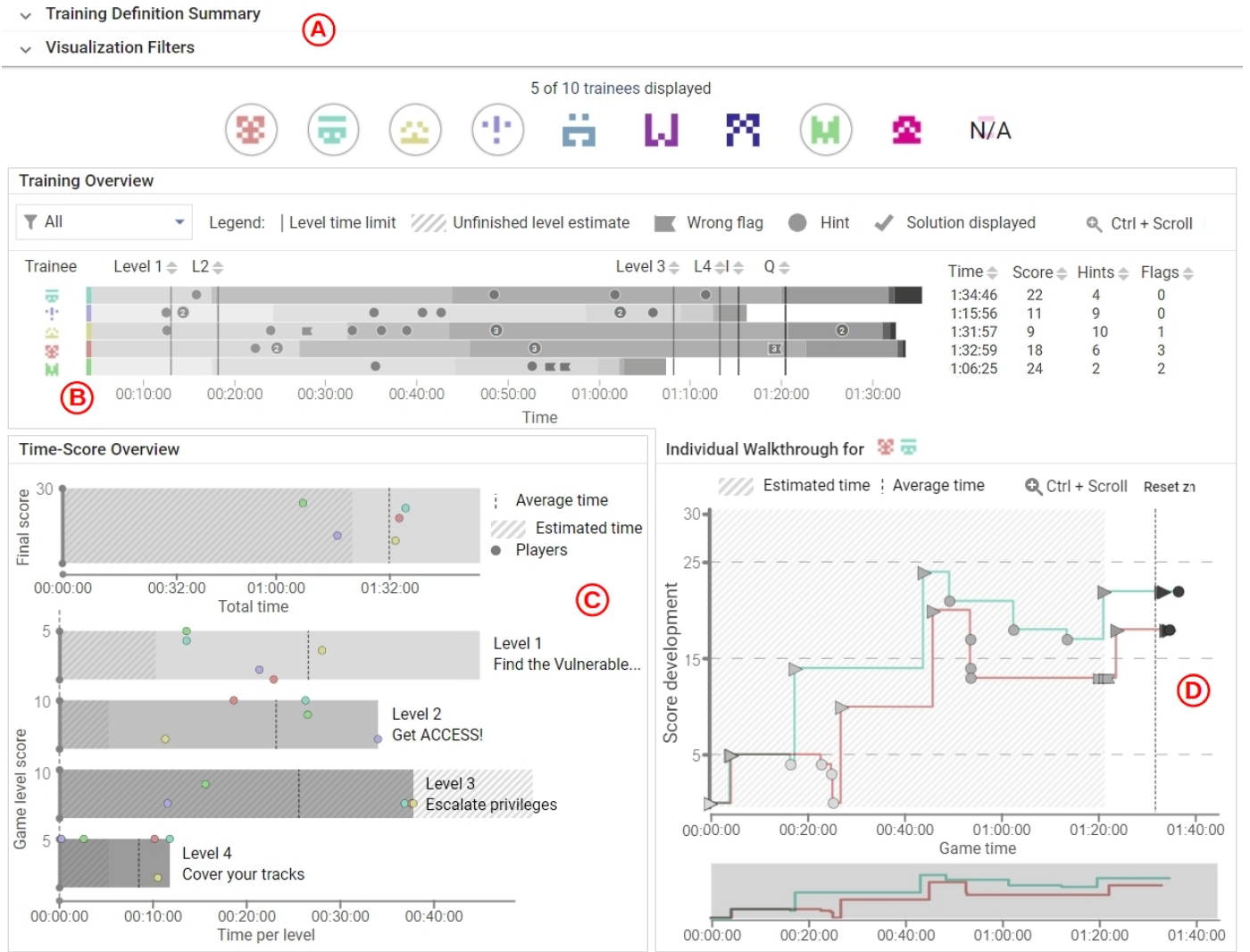
Fig. 5. The Training Analysis Tool (TAT) consists of the upper panel for training definition summary and filters (A) and three visualizations: the TRAINING OVERVIEW (B) displays the overall training duration for each trainee and their activities (e.g., taking hints, inserting incorrect flags). The TIME-SCORE OVERVIEW (C) presents the distribution of achieved scores (final and per-level) for each trainee. The INDIVIDUAL WALKTHROUGH (D) directly compares of two or three trainees and is subordinate to the TRAINING OVERVIEW.

for individual levels (Fig. 6 – A). For each game level, a table summarizing data of individual trainees provides an overview of the gained score, taken hints, incorrect flags, and time spent in the level (**R2** and **R3**). Comparing the results shown in the table with the level content and parameters (e.g., the comparison of incorrect flags with the correct flag or scheduled time allocation with the average or median values) can help the tutors identify problematic parts of the content (**R4**).

The VISUALIZATION FILTERS (Fig. 6 – B) are global filtering options to show or hide glyphs representing hints or flags and switch between trainees' avatars and names (IDs). The avatars (Fig. 6 – C) are switches for filtering out the trainees from the TRAINING OVERVIEW and TIME-SCORE OVERVIEW.

8.2. Training overview

We extended the TRAINING OVERVIEW (Fig. 5 – B) with the table summarizing total game duration, achieved score, number of taken hints, and submitted incorrect flags for each trainee. We also added the legend for quicker orientation.

The TRAINING OVERVIEW interacts with two complementary views. By clicking on the stacked bar, the INDIVIDUAL WALK-THROUGH visualization appears, showing score polyline and events of the corresponding trainee. The level bars highlight the corresponding dots in the TIME-SCORE OVERVIEW and the polyline in the INDIVIDUAL WALKTHROUGH on mouseover.

8.2.1. Time-score overview

Unlike the early prototype version, we added the dashed vertical line to indicate the actual average completion time of the trainees. The striped segments delimit the time estimate for each level. Therefore, the tutors can quickly identify the differences between the expected and the actual (and averaged) time for each level, as shown in Fig. 7.
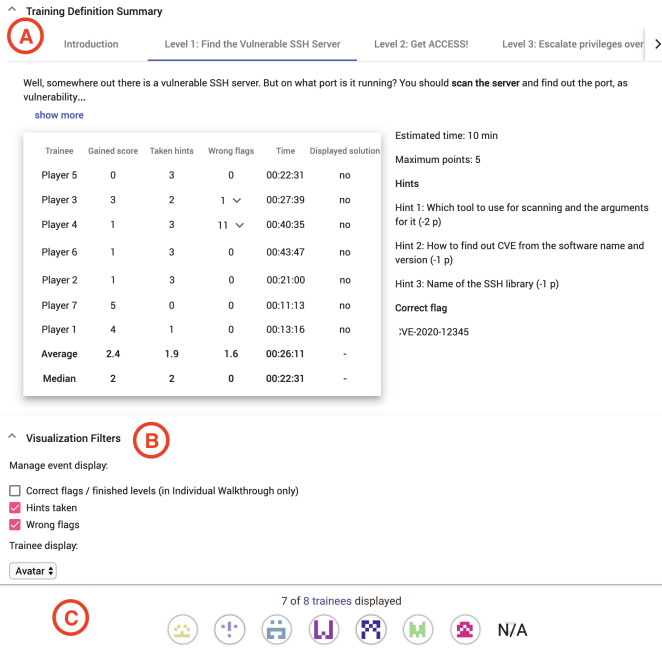
Fig. 6. TAT – details of the TRAINING DEFINITION SUMMARY (A), CONFIGURATION (B), and the TRAINEES (C) sections.
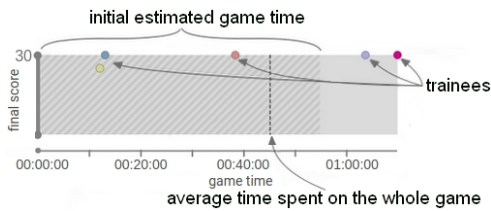

Fig. 7. The TIME-SCORE OVERVIEW combines bar charts with scatter plots to show relationships between the score and time of the game levels.

8.2.2. Individual walkthrough

In the final version, the INDIVIDUAL WALKTHROUGH (Fig. 5 – D) displays upon selecting a trainee in the TRAINING OVERVIEW. The selected trainees are indicated as avatars next to the title. We also reflected the main complaints regarding the clutteredness and simplified the visualization layout. Only the total training duration estimate is shown instead of the estimate for each level. We also added the vertical dashed line to indicate the actual average time, similarly to the TIME-SCORE OVERVIEW.

9. Summative evaluation

The summative evaluation was held in April 2020. We intended to validate the final design concerning the user requirements **R1–R4**, assess the usability and usefulness of the TAT, and identify possible refinements for the final integration into the *KYPO CRP*.

9.1. Participants

We asked the same six people who participated in the formative evaluation. We also recruited two more students who passed the CTF design course taught at our university (see Table 1). They represent novice users familiar with CTF games'

basic concepts and only have hands-on experience with their design.

9.2. Procedure

Due to the COVID-19 pandemic restrictions, we held it remotely using Google Meet, which we also used to record audio and screen. The participants used their computers or laptops with the 13.3"–27" screens and resolutions ranging from FullHD to UHD. The procedure was almost the same as in the formative evaluation (see Sec. 7.2). The only difference was a new data set that we used for the tasks.

DS3 uses data from a training session held as the introductory lecture of the CTF game design course of Fall 2019. It is an attack-oriented four-level training scenario similar to DS2, in this case, tested on nine trainees who generated 281 events in the session lasting 110 minutes. On average, each participant performed 31.3 events.

9.3. Results

The participants completed all the tasks without struggle. Despite minor difficulties, the immediate feedback was more positive than in the previous evaluation. Since the tasks are complex and depend on the tutor's knowledge and experience we sought qualitative input rather than measuring user performance.

Participants mostly worked with the TRAINING OVERVIEW since it contains most of the necessary information. The TIME-SCORE OVERVIEW serves well to identify timing issues and assess level difficulty. The TRAINING DEFINITION SUMMARY supports finding flaws in the puzzle assignments (e.g., misleading texts, wrong instructions for flag format). Further, we did an inductive qualitative analysis [43] of the video recordings, which is summarized below.

Visualizations usage. Figure 8 shows the usage of visualizations to solve the tasks by participants. The most preferred was the TRAINING OVERVIEW. All but P5 used the TRAINING OVERVIEW as a starting point when solving all the tasks (P5 preferred the TIME-SCORE OVERVIEW). Its most acclaimed feature is the ability to sort trainees by the time spent in individual levels and compare them to the estimated level duration (defined in the training scenario). Participants also used the visualization to identify the trainees who significantly exceeded the estimated level duration time. All the sorting options (by time spent in a level, final time, score, hints, and incorrect flags) were used at least once by each participant. On the other hand, the zooming function was used only rarely (P1, P6). The participants used the TIME-SCORE OVERVIEW to identify outlying trainees (P2, P4, P5), assess each level's difficulty based on their time/score distribution (P1, P2, P4, P5), or compare it with the maximum score per level (P3, P4). The INDIVIDUAL WALKTHROUGH was still considered the least usable (P1, P2, P5, P6, P7). P1 and P5 did not work with it at all. Others used it only for a direct comparison of two trainees (*T2*).

The TAT allows comparison of trainees beyond time and score. To identify non-standard trainees' behavior (*T1*), we observed that all the participants revealed all or almost all occurrences of the most common types, such as taking all hints at
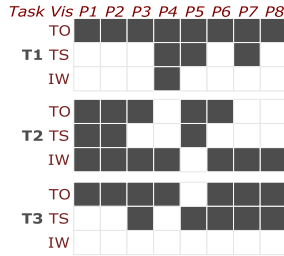
Fig. 8. Gray cells indicate visualization usage (Vis) when solving tasks T1–T3 for each participant (P1–P8). Visualizations: Training Overview (TO), Time-Score Overview (TS), Individual Walkthrough (IW).
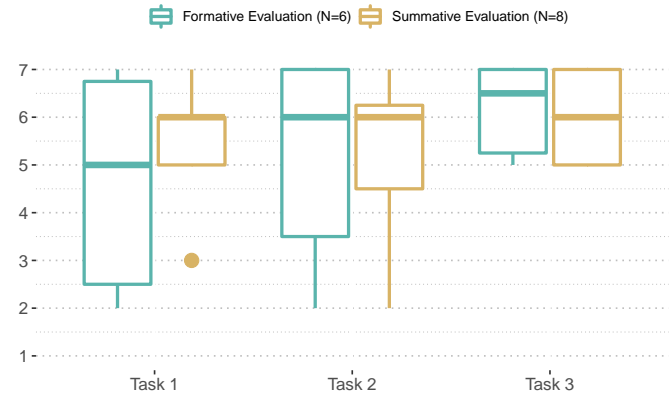

Fig. 9. Single Ease Question scores of the tasks in both evaluations.

Fig. 10. Summative evaluation: The SUS questionnaire responses.

once shortly after they entered a new level or guessing the flags in each dataset. The participants found those with the lowest score/largest time, followed by a detailed inspection of the number of taken hints and inserted incorrect flags. The procedure was the same for all. The difference was only in the starting visualization. While P4, P5, and P7 started with the TIME-SCORE OVERVIEW, the rest used the TRAINING OVERVIEW solely. The participants also intensively used the trainee filter combined with the TRAINING OVERVIEW sorting capabilities to filter out unwanted trainees quickly, especially for the second task (*T2*). Despite the INDIVIDUAL WALKTHROUGH received mixed reactions, most participants (except P1 and P5) used it for a head-to-head comparison.

The TAT helps to identify training scenario shortcomings. When dealing with the identification of training scenario shortcomings (*T3*), the participants mainly focused on three areas: correcting the time estimates and maximal score of individual levels, the perceived level difficulty, and instructions for a correct flag format. All the participants proposed changing the time estimates or the assigned maximum of points based on the trainees' overdue in the first two levels of D3. Moreover, seniors (P1, P3–P5) also identified the confusion with the flag formatting instructions in the second level. P3–P5 analyzed the data even more profoundly and revealed the flaw in the game design based on the observation that some trainees used the correct flag for the fourth level in the third one.

Except for P1, P2, and P4, the participants used the TRAINING DEFINITION SUMMARY since it clearly shows the difference between the estimate and real-time. The size of each level allows for a quick comparison of their perceived difficulty (the longer it took, the problematic the level was). The glyphs visualizing incorrect flags in the TRAINING OVERVIEW proved to be good indicators for potential issues with the puzzle assignments, including the technical instructions. All the experts (P1–P5) greeted the TRAINING DEFINITION SUMMARY as a convenient way to search for problematic parts of the training definition.

Gaps and drawbacks of the TAT. We received several suggestions for further improvements to the TAT visualizations. P5 suggested adding "the horizontal line also showing the average score per level" in the TIME-SCORE OVERVIEW to improve comparing level scores. The two-level filtering (avatars → trainees in the TRAINING OVERVIEW) received mixed feedback. Only three participants (P1, P2, P5) used both to filter out specific trainees, while others preferred to keep all of them visible. The evaluation also revealed that with the grayscale for the TRAINING
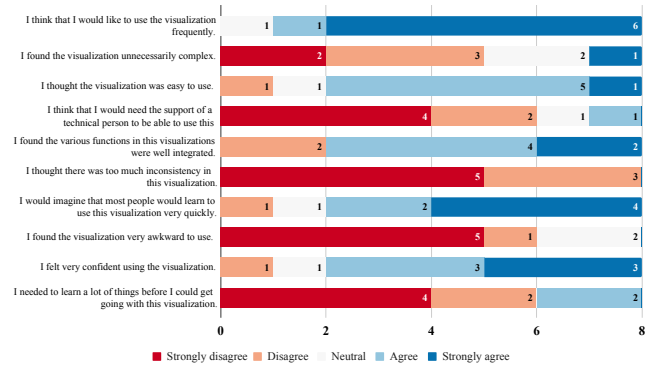
OVERVIEW, highlighting of selected trainees is not very pronounced and will be revised in future development.

The main benefit of the INDIVIDUAL WALKTHROUGH is that the polyline visualizing score development better informs the tutor whether there are similarities in the trainees' gameplay. Since this is useful only in a specific use case, we will reconsider its integration in the subsequent design iterations simplifying the user interface.

The average SUS score raised to 77.5 (compared to 65.4 for the early prototype), which still equals to *good* rating. We assume that it is mainly due to the higher complexity of the tool and the remaining issues with the INDIVIDUAL WALKTHROUGH. The data plot of the SUS questionnaire responses is in Fig. 10. However, the medians 6.0 of SEQ score (Fig. 9) for all the tasks (*T1–T3*) further supports our statement that the TAT is well-suited for the post-training analysis.

10. Discussion

In this section, we discuss the findings and limitations of the studies.

10.1. Lessons learned

The summative evaluation validated our design decisions. The verbal feedback from participants and the SEQ and SUS

scores confirmed that the tools address the elicited requirements. We also revealed three notable findings regarding the presentation of summaries, sorting and filtering capabilities, and domain specificity.

Summaries. Extending the visualization with pertinent summary data could help tutors to overview the situation and identify anomalies quickly. Especially in analytical tools, even elementary statistics and simple charts are helpful. Although we did not implement such charts in the TAT, some participants asked for them as feature requests.

Sorting and filtering. The evaluation revealed that we should work with the sorting and filtering options even more thoughtfully so that tutors can better focus their attention. There must be a real usage scenario for each filter type. Particular attention should be paid to carefully selecting items for filtering and the batch selecting and filtering shortcuts (e.g., "deselect all").

Domain-specific insight over universality and scalability. Puzzle-based learning represents a vast area where tutors' support tools differ vastly among various application domains. Since there are no guidelines or best practices and the user requirements are often contradictory, they have to be considered carefully, and the tools should be tailored to specific uses. Furthermore, the amount of data from a single session is usually relatively small.

10.2. Limitations

Both user studies had two main limitations to the external validity: low number of participants and qualitative focus of the evaluation in the controlled environment instead of the in-the-wild evaluation.

To ensure the evaluation's ecological validity, we needed users with practical experience with organizing hands-on training sessions and knowledge of cybersecurity education. These demands notably restrict our choice of suitable candidates. Our collaborating cybersecurity educators are, no doubt, the primary users of the developed tools. Therefore, they provided relevant feedback, which will serve as a source for further thoughts on both tools' improvements. We also asked students of the cybersecurity degree program who successfully passed the university course on CTF games design. They represent novice users unfamiliar with analytical visualizations.

Due to the qualitative nature of the evaluations, we did not focus on finding the limits in terms of the total number of trainees and their events since the events with more than 16 participants are literally none due to the space limits of the training facility at our university. We originally planned to perform the case studies to assess the TAT's final design in the actual deployment. Unfortunately, due to the COVID-19 pandemic, the scheduled hands-on training sessions had been canceled, and the only feasible option was to perform the evaluation remotely, using the same procedure as in the summative evaluation.

In this work, we restrict ourselves to the case study of hands-on cybersecurity courses focused on system hacking and cyber-attacks. In particular, puzzle-based capture the flag games where the structure and data are well-defined in advance. These restrictions allowed us to provide the tutors with a more in-depth insight into this specific application sub-domain through a pair of visualization tools.

Despite these limitations, the provided feedback has been guiding our work and feature requests for the deployment into the *KYPO CRP*.

11. Conclusion and future work

We introduced the visual analytics tool that, based on the qualitative feedback, improves the tutors' insight into the training sessions and allows them to assess the quality of the training scenarios and evaluate the training session results. We focused on low-level learning analysis (i.e., analyzing data from a single training session). As we pointed out in Sec. 2, this particular area is often overlooked since the main focus in support tools for tutors and educators is on high-level analysis for MOOC e-learning.

We have presented a design study on applying visual analytics to data from hands-on cybersecurity training in the form of CTF games. We introduced two iterations of the *Training Analysis Tool*, allowing tutors to assess the quality of the training scenarios and gain insight into the trainees' progress beyond the completion time and final score. The summative evaluation validated our design decisions. The verbal feedback from participants and the SEQ and SUS scores confirmed that the tools address the elicited requirements. We gradually learned more about what information tutors would like to display in the visualization and how they interact with the data during the design study. Based on this experience, we believe that a data-driven insight into the training courses could provide surprising insights and knowledge about the design and behavior of trainees.

Focusing on puzzle-games principles enabled us to conceptualize the data and visualizations beyond the cybersecurity domain. If we look closely at the information we used, we realize that it is a quadruple: timestamp, the ID of the trainee, type of event, content (arbitrary). Therefore, we believe that our approach can be easily applied in other areas where hands-on training becomes common. We admit that there are further requirements, such as automated processing of user inputs, but even basic logging can provide sufficient data. The level of detail depends mainly on the expressiveness of the content component.

Consider the university programming course as another application area. The tutors often evaluate students' assignments using automated compilation and validation tools against predefined unit tests and datasets. The summary of code diffs, compiler error logs, and output of the automated tests can be logged. Similar to the cybersecurity domain, these events can be mapped to assessment events (e.g., penalties for unsuccessful unit tests), player actions (e.g., the submission of a piece of code), and progress events (e.g., successful compilation and test of a programming task). Visualizing these events on the timelines (one per student) or further text analysis of the code can be as valuable as our analogy with the cybersecurity CTF games.

The support tools for a category of so-called blended classrooms and hands-on courses are still mostly unexplored. Our

work addresses only a tiny part of this broad research area. Despite our focus on cybersecurity education, we consider our findings applicable in other areas of puzzle-based learning and analyzing data from a single training session (i.e., low-level learning analysis). We want to encourage others to explore novel methods for visual analysis of puzzle-based learning courses in different areas.

The TAT is integrated into the user interface of the *KYPO CRP*. We also work on additional data integration from sandboxes (e.g., resource usage, executed commands, running processes). Enhancing the current level of event processing with this information will further improve the insight and enable a more detailed analysis of the training and its scenario. Our next goal is to explore the possibilities for visual analysis of multiple training sessions and analyze and assess trainees' long-term progress. Extending the analysis with automatic highlighting of anomalies or flaws in the training design is another direction of research that needs further study.

References

[1] Medeiros, RP, Ramalho, GL, Falcão, TP. A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. IEEE Trans on Education 2018;62(2):77–90.

[2] McMurtrey, ME, Downey, JP, Zeltmann, SM, Friedman, WH. Critical Skill Sets of Entry-level IT Professionals: An Empirical Examination of Perceptions from Field Personnel. J of Inf Tech Education: Research 2008;7:101–120.

[3] Michalewicz, Z, Michalewicz, M. Puzzle-based learning. Ormond, Australia: Hybrid Publishers; 2008.

[4] Yoneyama, Y, Matsushita, K, Mackin, KJ, Ohshiro, M, Yamasaki, K, Nunohiro, E. Puzzle Based Programming Learning Support System with Learning History Management. In: Proc. of the 16th Int. Conf. on Computers in Education. New York: IEEE Press; 2008, p. 623–627.

[5] Merrick, KE. An Empirical Evaluation of Puzzle-based Learning as an Interest Approach for Teaching Introductory Comp. Science. IEEE Trans on Education 2010;53(4):677–680.

[6] Harms, KJ, Rowlett, N, Kelleher, C. Enabling Independent Learning of Programming Concepts Through Programming Completion Puzzles. In: 2015 IEEE Symp. on Visual Languages and Human-Centric Computing (VL/HCC). IEEE; New York: IEEE Press.; 2015, p. 271–279.

[7] Gondree, M, Peterson, ZN, Denning, T. Security Through Play. IEEE Security & Privacy 2013;11(3):64–67.

[8] Hendrix, M, Al-Sherbaz, A, Victoria, B. Game-based Cyber Security Training: Are Serious Games Suitable for Cyber Security Training? Int J of Serious Games 2016;3(1):53–61.

[9] Dasgupta, D, Ferebee, DM, Michalewicz, Z. Applying Puzzle-Based Learning to Cyber-Security Education. In: Proc. of the 2013 on InfoSecCD '13: Information Security Curriculum Development Conf. New York, NY, USA: ACM; 2013, p. 20:20–20:26.

[10] Loh, CS. Information trails: In-process assessment of game-based learning. In: Assessment in game-based learning. Berlin, Heidelberg: Springer; 2012, p. 123–144.

[11] Chung, GK. Guidelines for the design and implementation of game telemetry for serious games analytics. In: Serious games analytics. Berlin, Heidelberg: Springer; 2015, p. 59–79.

[12] Alonso-Fernandez, C, Calvo, A, Freire, M, Martinez-Ortiz, I, Fernandez-Manjon, B. Systematizing game learning analytics for serious games. In: 2017 IEEE global engineering education Conf. (EDUCON). IEEE; New York: IEEE; 2017, p. 1111–1118.

[13] Owen, VE, Baker, RS. Fueling prediction of player decisions: Foundations of feature engineering for optimized behavior modeling in serious games. Technology, Knowledge and Learning 2020;25(2):225–250.

[14] Ošlejšek, R, Rusňák, V, Burská, K, Švábenský, V, Vykopal, J, Čegan, J. Conceptual model of visual analytics for hands-on cybersecurity training. IEEE Trans on Vis and Comp Graphics 2020;:1–13.

[15] Rodríguez-Triana, M, Prieto, L, et al. Monitoring, Awareness and Reflection in Blended Technology Enhanced Learning: a Systematic Review. Int J of Tech Enhanced Learning 2016;9.

[16] Gratzl, S, Lex, A, Gehlenborg, N, Pfister, H, Streit, M. Lineup: Visual analysis of multi-attribute rankings. IEEE transactions on visualization and computer graphics 2013;19(12):2277–2286.

[17] Matcha, W, Gašević, D, Uzir, NA, Jovanović, J, Pardo, A. Analytics of Learning Strategies: Associations with Academic Performance and Feedback. In: Proc. of the 9th Int. Conf. on Learning Analytics & Knowledge. New York, NY, USA: ACM; 2019, p. 461–470.

[18] Jivet, I, Scheffel, M, Specht, M, Drachsler, H. License to Evaluate: Preparing Learning Analytics Dashboards for Educational Practice. In: Proc. of the 8th Int. Conf. on Learning Analytics and Knowledge. New York, NY, USA: ACM; 2018, p. 31–40.

[19] Ošlejšek, R, Vykopal, J, Burská, K, Rusňák, V. Evaluation of cyber defense exercises using visual analytics process. In: 2018 IEEE Frontiers in Education Conference (FIE). IEEE; 2018, p. 1–9.

[20] de Freitas, S, Gibson, D, et al. How to Use Gamified Dashboards and Learning Analytics for Providing Immediate Student Feedback and Performance Tracking in Higher Education. In: Proc. of the 26th Int. Conf. on World Wide Web Companion. Geneva, Switzerland: Int. World Wide Web Conf.s Steering Committee; 2017, p. 429–434.

[21] Loh, CS, Sheng, Y, Ifenthaler, D. Serious Games Analytics: Methodologies for Performance Measurement, Assessment, and Improvement. Springer International Publishing; 2015.

[22] Macfadyen, LP, Dawson, S. Mining LMS Data to Develop an "Early Warning System" for Educators: A Proof of Concept. Computers & Education 2010;54(2):588–599.

[23] Siemens, G, Long, P. Penetrating the Fog: Analytics in Learning and Education. EDUCAUSE Review 2011;46(5):30.

[24] Verbert, K, Govaerts, S, Duval, E, Santos, JL, Van Assche, F, Parra, G, et al. Learning Dashboards: An Overview and Future Research Opportunities. Personal and Ubiquitous Computing 2013;18:1499–1514.

[25] Verbert, K, Duval, E, Klerkx, J, Govaerts, S, Santos, JL. Learning Analytics Dashboard Applications. Am Behavioral Scientist 2013;57(10):1500–1509.

[26] Schwendimann, BA, Rodríguez-Triana, MJ, Vozniuk, A, Prieto, LP, Boroujeni, MS, Holzer, A, et al. Perceiving Learning at a Glance: A Systematic Literature Review of Learning Dashboard Research. IEEE Trans on Learning Tech 2017;10(1):30–41.

[27] Bodily, R, Verbert, K. Review of Research on Student-Facing Learning Analytics Dashboards and Educational Recommender Systems. IEEE Trans on Learning Tech 2017;10(4):405–418.

[28] Sébastien, V, Sébastien, D, Timol, I, Gay, D, Cucchi, A, Porlier, C. Moodleboard: Dynamic and Interactive Indicators for Teachers and Pedagogical Engineers. In: 2019 Conf. on Next Generation Computing Applications (NextComp). New York: IEEE; 2019, p. 1–5.

[29] Charleer, S, Moere, AV, Klerkx, J, Verbert, K, De Laet, T. Learning Analytics Dashboards to Support Adviser-Student Dialogue. IEEE Trans on Learning Tech 2018;11(3):389–399.

[30] Govaerts, S, Verbert, K, Klerkx, J, Duval, E. Visualizing Activities for Self-reflection and Awareness. In: Int. Conf. on Web-based Learning. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010, p. 91–100.

[31] Jacobs, KL. Investigation of Interactive Online Visual Tools for the Learning of Mathematics. Int J of Mathematical Education in Sci and Technology 2005;36(7):761–768.

[32] Fu, X, Shimada, A, Ogata, H, Taniguchi, Y, Suehiro, D. Real-time Learning Analytics for C Programming Language Courses. In: Proc. of the Seventh Int. Learning Analytics & Knowledge Conf. New York, NY, USA: ACM; 2017, p. 280–288.

[33] Falah, A, Pan, L, Abdelrazek, M. Visual Representation of Penetration Testing Actions and Skills in a Technical Tree Model. In: Proc. of the Australasian Comp. Sci. Week Multiconference. New York, NY, USA: ACM; 2017, p. 8:1–8:10.

[34] Werther, J, Zhivich, M, Leek, T, Zeldovich, N. Experiences in Cyber Security Education: The MIT Lincoln Laboratory Capture-the-flag Exercise. In: Proc. of the 4th Conf. on Cyber Security Experimentation and Test. Berkeley, CA, USA: USENIX Association; 2011, p. 1–12.

[35] Davis, A, Leek, T, Zhivich, M, Gwinnup, K, Leonard, W. The Fun and Future of CTF. In: 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14). San Diego, CA: USENIX Association; 2014, p. 1–9.

[36] Švábenský, V, Vykopal, J, Cermak, M, Laštovička, M. Enhancing Cybersecurity Skills by Creating Serious Games. In: Proc. of the 23rd Annual ACM Conf. on Innovation and Tech. in Comp. Sci. Education. ACM; New York, NY, USA: ACM; 2018, p. 194–199.

[37] Čeleda, P, Čegan, J, Vykopal, J, Tovarňák, D. KYPO – A Platform for Cyber Defence Exercises. In: STO-MP-MSG-133: M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence. Munich (Germany): NATO Science and Technology Organization; 2015, p. 12.

[38] Sedlmair, M, Meyer, M, Munzner, T. Design Study Methodology: Reflections from the Trenches and the Stacks. IEEE Transactions on Visand Comp Graphics 2012;18(12):2431–2440.

[39] Ošlejšek, R, Rusňák, V, Burská, K, Švábenský, V, Vykopal, J. Visual Feedback for Players of Multi-Level Capture the Flag Games: Field Usability Study. In: Proc. of the IEEE Symp. on Vis.for Comp. Security (VizSEC). New York: IEEE Press.; 2019, p. 1–11.

[40] Sauro, J. A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices. USA: CreateSpace Independent Publishing Platform; 2011.

[41] Sauro, J, Dumas, JS. Comparison of Three One-question, Post-task Usability Questionnaires. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York, NY, USA: ACM; 2009, p. 1599–1608.

[42] Bangor, A, Kortum, P, Miller, J. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. J Usability Studies 2009;4(3):114–123.

[43] Thomas, DR. A General Inductive Approach for Analyzing Qualitative Evaluation Data. Am J of Evaluation 2006;27(2):237–246.